**UNIVERSIDAD AUTÓNOMA DE MADRID**

ESCUELA POLITÉCNICA SUPERIOR



**TRABAJO FIN DE MÁSTER**

# Control y gestión de sondas de monitorización Ethernet usando NETCONF y modelos de datos YANG

**Máster Universitario en Ingeniería de Telecomunicación**

**Autor: CUCHARERO ATIENZA, Tito**
**Tutor: RAMOS DE SANTIAGO, Javier**
**Ponente: LÓPEZ DE VERGARA MÉNDEZ, Jorge E.**

**FECHA: Septiembre, 2017**

# Control y gestión de sondas de monitorización Ethernet usando NETCONF y modelos de datos YANG

**AUTOR: Tito Cucharero Atienza**
**TUTOR: Javier Ramos de Santiago**
**PONENTE: Jorge E. López de Vergara Méndez**

**Grupo HPCN**
**Dpto. Tecnología Electrónica y de las Comunicaciones**
**Escuela Politécnica Superior**
**Universidad Autónoma de Madrid**
**Septiembre de 2017**

# Resumen

Actualmente la mayoría de dispositivos electrónicos que utilizamos están conectados a Internet. Para que todos estos dispositivos tengan un buen rendimiento se necesita una buena calidad de red. La monitorización de parámetros de calidad de servicio (QoS) nos permite detectar anomalías y conocer por qué se están produciendo o dónde están los problemas. Con el paso del tiempo la complejidad de los sistemas de monitorización ha crecido y la gestión de los mismos (realizada tradicionalmente con protocolos como SNMP) se ha convertido en una tarea muy importante. Por este motivo, es necesario crear sistemas de gestión y control de sondas que sean estándares, interoperables y sencillos. En concreto, el protocolo estándar NETCONF junto con las definiciones de modelos de datos en YANG están siendo utilizados en los últimos años para gestionar de manera transparente equipos de red y middleboxes.

En este proyecto se aborda el estudio y la posterior implementación del control remoto de sondas Ethernet, a través del protocolo NETCONF y su modelo de datos YANG asociado. El protocolo NETCONF define un mecanismo simple a través del cual podemos gestionar un dispositivo de red, obtener información sobre la configuración del dispositivo y manipular o actualizar ficheros de configuración. YANG es un lenguaje de modelado de datos que define una jerarquía sobre los mismos. Además, se propone la implementación de estas funcionalidades integradas dentro de un entorno Web que permita realizar operaciones fácilmente y de una manera práctica para el usuario.

En concreto, en este proyecto se ha implementado la ejecución remota de varias herramientas de análisis de red (tanto activas como pasivas) con las que se puede obtener una visión global del rendimiento de la misma. Esta característica dota al entorno web desarrollado de la posibilidad de analizar datos en tiempo real y no depender de la propia configuración de la sonda, permitiendo a los analistas de red realizar mediciones en momentos en los que se detecte alguna anomalía. La implementación propuesta en este trabajo ha sido probada con éxito en un entorno virtual que ha permitido validar el correcto funcionamiento de la herramienta, así como definir los casos de uso de utilidad para los analistas y gestores de red.



# Palabras clave

NETCONG, YANG, herramientas de análisis de red, medidas de red, monitorización.

# Abstract

Currently most of the electronic devices are connect to the Internet. Good network quality is required for all these devices. Monitoring Quality of Service (QoS) parameters allows us to detect anomalies and know why they are occurring. Over time, the complexity of monitoring systems has grown and management (traditionally performed with protocols such as SNMP) has become a very important task. For such a reason, it is necessary to create management and control systems for probes that are standard, interoperable and simple. Specifically, the standard protocol NETCONF along with the definitions of YANG data models are being used in recent years to manage network equipment and middleboxes.

In this project we will study and implement the remote control of Ethernet probes, using the NETCONF protocol and YANG data models. The NETCONF protocol defines a simple mechanism through which we can manage a network device, obtain information about the configuration of the device and manipulate or update configuration files. YANG is a data modeling language that defines a hierarchy over them.
In addition, this work proposes the implementation of these functionalities integrated within a Web environment that allows perform operations easily.

In particular, this project has implemented the remote execution of several network analysis tools (for active and passive measurements) that allows obtaining an overall view of the performance of the network This feature provides the developed web environment the possibility of analyzing data in real time and not depending on the probe's own configuration, thus allowing network analysts to perform measurements when an anomaly is detected. The implementation proposed in this work has been successfully tested in a virtual environment that has allowed validating the correct operation of the tool, as well as defining useful use cases for analysts and network managers.

# Key Words

NETCONG, YANG, network analysis tools, network measurements, monitoring.

## *Agradecimientos*

*Quiero agradecer a mi familia y a mi novia por su apoyo constante.*

*A Javier Ramos, tutor de este trabajo de fin de master, por su paciencia y dedicación, haciendo posible la realización de este trabajo.*

# ÍNDICE DE CONTENIDOS

# ÍNDICE DE FIGURAS

# ÍNDICE DE TABLAS

# 1 Introducción

En este capítulo se presenta la motivación y objetivos que han sido necesarios para la realización de este Trabajo de Final de Máster. A continuación, hablaremos del porqué de su realización, así como sus principales objetivos. Concluiremos el capítulo esquematizando la estructura de este documento.

## *1.1 Introducción*

En la actualidad, el protocolo simple de gestión de red (Simple Network Management Protocol, SNMP) [1] es ampliamente usado para la obtención de estadísticas y monitorización de las redes de comunicaciones. Con el paso del tiempo la gestión de los dispositivos de red se ha vuelto muy compleja y las funcionalidades de SNMP empiezan a resultar insuficientes dada la variedad de dispositivos y tipos de redes. En este escenario, el protocolo NETCONF [2] ha cubierto las carencias de SNMP añadiendo mecanismos genéricos y extensibles para la configuración y gestión de equipos. NETCONF está presente en la actualidad en varios dispositivos de red como pueden ser routers y switches. La principal funcionalidad de este protocolo es la obtención de la configuración del dispositivo y su posterior modificación si se desea. Siguiendo la tendencia actual de la industria, en este trabajo se propone realizar la gestión y configuración de sondas Ethernet dedicadas a medidas de red utilizando NETCONF.

Debido a la importancia de los sistemas de monitorización, se propone desarrollar funcionalidades que permitan configurar sondas de monitorización Ethernet, así como orquestar medidas y obtener resultados. Todo ello se hará mediante el uso de un entorno web con el objetivo de convertirlo en una herramienta de uso diario que facilite la labor de los analistas y gestores de red.

## *1.2 Motivación*

Actualmente la mayoría de dispositivos electrónicos que utilizamos están conectados a Internet. Para que todos estos dispositivos tengan un buen rendimiento se necesita una buena calidad de red. La monitorización de red se encarga de analizar el estado de la misma midiendo sus características a través de ciertos parámetros. Dichos parámetros se conocen como parámetros de calidad de servicios (QoS). La monitorización nos permite obtener informes de la calidad de la red a la que nos conectamos, también nos proporciona herramientas para poder detectar anomalías y conocer por qué se están produciendo o localizar donde están los problemas. Generalmente, si los sistemas de monitorización son suficientemente complejos e incluyen numerosas sondas, su gestión y control se complica de sobremanera. En este sentido, operaciones tan simples como activar o desactivar módulos de medida concretos o verificar el buen funcionamiento de una sonda concreta se vuelven muy complejas. Además, si las sondas son desarrollos propietarios se suele requerir el uso de protocolos y soluciones cerradas e interoperables. Por esto motivo, es necesario crear sistemas de gestión y control de sondas que sean estándares, interoperables y sencillos. En concreto, el protocolo estándar NETCONF junto con las definiciones de modelos de datos en YANG están siendo utilizados en los últimos años para gestionar de

manera transparente equipos de red y middleboxes. En esta línea surge la necesidad de crear sistemas de gestión y control de sondas de monitorización basadas en estas tecnologías.

Este TFM está relacionado con las asignaturas: Gestión de Redes, Planificación de Redes, Tecnologías y Servicios de Telecomunicación y Proyectos en Ingeniería de Telecomunicación. Entre dichas asignaturas, existe un mayor vínculo con Gestión de Redes, en la que se analizó el uso del protocolo SNMP. En el mismo contexto está el protocolo NETCONF. Por ello, en este trabajo se aplicarán los conocimientos adquiridos para profundizar más en el manejo de dispositivos de red y modelado de datos, con la motivación final de conseguir los objetivos citados.

## *1.3 Objetivos*

El principal objetivo de este TFM consiste en la creación, despliegue y configuración de sondas de monitorización Ethernet utilizando el protocolo estándar NETCONF. Para ello se desarrollará un entorno web que permita enviar órdenes a las sondas de monitorización situadas en la red. Estas órdenes se proporcionarán mediante el uso del protocolo NETCONF. NETCONF proporciona los mecanismos para instalar, configurar, manipular y eliminar dispositivos de red. Además, NETCONF permite la ejecución de llamadas a Procedimientos Remotos (*Remote Procedure Calls*, RPCs) [4]. A través de dichas RPCs podemos ejecutar scripts y programas en los dispositivos remotos de monitorización y recoger los resultados en un formato sencillo y definido. Como modelo de datos asociado a las RPCs se hará uso de YANG [5] para definir las operaciones, así como los datos de entrada y salida de cada una. En definitiva, los objetivos que plantea este TFM son:

- Desarrollo de un entorno web que permita visualizar y controlar sondas de monitorización.
- Creación de modelos YANG que representen operaciones, configuraciones y datos de medida que pueden ser obtenidos en las sondas Ethernet.
- Implementación de una capa de middleware que permita asociar RPCs con acciones específicas de monitorización.

El entorno web debe mostrar la información de la manera más intuitiva posible, pues uno de los objetivos es que sea una herramienta usable y de fácil acceso. En el propio entorno se ofrecerá la posibilidad a los usuarios de visualizar parámetros de calidad de servicio obtenidos en tiempo real. También se debe poder consultar un histórico de mediciones realizadas anteriormente para detectar posibles incidencias.

## *1.4 Organización de la memoria*

Este trabajo de fin de máster sigue la organización que se describe a continuación:

En el capítulo 1, se realiza un resumen introductorio al trabajo realizado, dando a conocer al lector los pilares del TFM. Se explica cuál ha sido el motivo de la realización de este proyecto, así como sus objetivos principales. A continuación, en el capítulo 2, se exponen los distintos mecanismos de calidad de servicio. También realizaremos una descripción de los métodos y tipos de medida utilizados, así como las principales herramientas de análisis de red utilizadas en este trabajo. Concluiremos con una comparativa respecto a otros trabajos realizados. Después, en el capítulo 3, analizaremos el diseño llevado a cabo separándolo en tres partes: diseño web, aplicaciones NETCONF y diagrama de caso de uso. En el capítulo 4, explicaremos como ha sido el desarrollo de este proyecto y visualizaremos el resultado final desde el entorno web. Por último, en el capítulo 5, se muestran las conclusiones más relevantes de este trabajo, así como las futuras líneas de trabajo que pueden seguirse para ampliarlo.

# 2 Estado del arte

En este capítulo se explicarán de manera resumida los fundamentos necesarios para la comprensión y realización de este trabajo. En concreto, se abordarán las metodologías, tecnologías y protocolos más relevantes y se describirá el funcionamiento de las medidas activas y pasivas y sus principales métodos [7]. Por último, se hará una comparativa con otros trabajos realizados que abordan temas y funciones semejantes.

## *2.1 Parámetros de QoS (Quality of Service)*

La calidad de servicio es definida por la Unión Internacional de Telecomunicaciones (UIT) "*como el efecto global de la calidad de funcionamiento de un servicio que determina el grado de satisfacción de un usuario de dicho servicio*". A fin de obtener los parámetros de calidad de servicio en redes domésticas y redes comerciales, los operadores aplican diferentes técnicas de control de tráfico en routers intermedios. Estas técnicas pueden producir efectos indeseables como la pérdida de paquetes o la disminución de la velocidad de los flujos de paquetes que están atravesando la red. En este escenario la comprensión de los mecanismos QoS resulta ser de suma importancia en el diseño e implementación de técnicas de medida. Para poder asegurar el correcto funcionamiento de los servicios y aplicaciones que se proveen a través de la red, es necesario monitorizar un conjunto de parámetros.

### Parámetros relevantes de QoS:

- **Capacidad**

La capacidad en un enlace de nivel dos se define como la velocidad de transmisión constante. Tal velocidad de transmisión invariable está limitada por las características físicas del medio de transmisión y por las características ópticas /eléctricas del hardware del transmisor y receptor. En el nivel IP (Internet Protocol) [8], la capacidad se entiende como la velocidad de transmisión teniendo en cuenta la sobrecarga producida por las cabeceras de la capa de enlace.

- **Ancho de banda disponible**

El ancho de banda disponible de un camino extremo a extremo es la capacidad no usada en un periodo de tiempo dado. Esta métrica depende tanto de las características físicas como de la carga del enlace a lo largo del tiempo. Para calcular el ancho de banda disponible es necesario conocer la carga del enlace con antelación, lo cual es complicado. Generalmente se realizan estimaciones en periodos temporales cortos (por ejemplo, cinco minutos) para contar con un valor realista de carga.

- **Throughput**

Otro parámetro de medida importante relacionado con el ancho de banda es el throughput o rendimiento de una conexión TCP. La principal desventaja de este parámetro es su dependencia con diferentes factores tales como el número de conexiones TCP concurrentes, el tamaño de la transferencia de datos, la congestión de los enlaces o la cantidad de tráfico cruzado presente en el enlace.
El rendimiento de una conexión TCP se conoce también con el término Bulk Data Transfer Capacity (BTC) [9].
Hay que tener en cuenta que el throughput es una métrica no aplicable en todos los escenarios debido a la dependencia con otros parámetros.

- **Retardo en un sentido (One-Way Delay [OWD])**

El retardo en un sentido se puede definir como el tiempo transcurrido entre el primer bit de un paquete en un punto de observación origen y el último bit del mismo paquete en un punto de observación destino [10]. El retardo en un sentido se compone de: el retardo de transmisión, el retardo de propagación, el retardo de procesamiento y el retardo de encolado en los equipamientos intermedios de la red [11]. El retardo de transmisión es el tiempo requerido para transmitir todos los bits de un paquete dado. Este retardo depende de la longitud del paquete, de la tasa de transmisión y del medio físico. El retardo de propagación es el tiempo transcurrido desde que se emite el último bit de un paquete hasta que ese mismo bit es recibido en el destino. El retardo de procesado es el tiempo que necesita cada router o equipo de red para procesar un paquete. Y, por último, el retardo de encolado es el tiempo gastado en la cola de un router o equipamiento de red hasta que es procesado.

- **Retardo ida y vuelta (Round-Trip Time [RTT])**

El retardo ida y vuelta se define como el intervalo de tiempo entre el primer bit del envío de un segmento TCP y el último bit del correspondiente paquete ACK de TCP recibido [12]. Aunque el RTT está definido sobre TCP, el concepto puede ser extendido a cualquier protocolo bidireccional [13]. A diferencia del retardo en un sentido, el RTT proporciona información sobre las dos direcciones de la comunicación, lo cual es útil cuando se realizan medidas en enlaces asimétricos, como, por ejemplo, las líneas de abonado digital (Digital Subscriber Line [DSL]). El RTT también toma en cuenta el tiempo de procesamiento en cada extremo de la conexión. Por ejemplo, la medida del RTT de una conexión que utiliza HTTP implica el tiempo de espera que necesita un servidor para generar una respuesta HTTP o un paquete TCP RST en caso de que la conexión no pueda realizarse. La medida del RTT en este escenario implica incluir el retardo de procesamiento del servidor HTTP como parte del retardo de comunicaciones, lo cual puede no ser aceptable en algunos casos.

- **Jitter**

El término jitter resulta, en muchas ocasiones, mal usado dependiendo del contexto. En el escenario de medida de parámetros de QoS, el término jitter se refiere a la variación del retardo de los paquetes de un flujo dado. A partir de ahora, se va a utilizar el término variación del retardo de paquetes en lugar de jitter. La variación del retardo en un flujo de paquetes puede ser definida como la diferencia entre el retardo en un sentido de un grupo de paquete determinados [14]. Los paquetes pueden seleccionarse por medio de una función de selección determinista o aleatoria aplicada al conjunto total de paquetes recibidos. Para el cálculo de la variación del retardo se deben utilizar únicamente pares de paquetes ordenados. Otro enfoque define la variación del retardo de paquetes como la desviación estándar del retardo en un sentido de los paquetes observados en un periodo de tiempo determinado [15]. El cálculo de la variación del retardo debe realizarse eliminando las muestras en la que se observan pérdidas como se indica en [10]. Además del método anteriormente comentado, se puede recurrir al cálculo del Coeficiente de Variación (CV) como medida de la variación del retardo en un sentido. Este enfoque provee una medida normalizada de la dispersión del retardo en un sentido. A diferencia de los enfoques anteriores, el método de cálculo del CV ofrece una magnitud sin unidades que da una idea acerca de si la variación del retardo es grande o no. Cuanto mayor sea el coeficiente de variación, mayor será la variabilidad de retardo en un sentido. Esta aproximación resulta útil cuando la información relativa se utiliza para determinar la calidad de un camino extremo a extremo.

- **Pérdidas de paquetes**

La pérdida de paquetes es un parámetro crucial en el análisis de calidad de servicio. Ésta se define como la cantidad de paquetes perdidos respecto del total de paquetes enviados en un periodo temporal determinado [16]. Un paquete se considera perdido si no llega a su destino, llega con errores o se recibe con un retardo excesivo. Hay que tener en cuenta que un paquete puede no llegar a su destino por diversas causas como pueden ser: paquetes descartados en una cola de un equipamiento a lo largo de un camino extremo a extremo o, incluso, problemas físicos en enlaces. En el caso de protocolos que permiten la fragmentación como IP, un paquete se considera como perdido si al menos uno de sus fragmentos se pierde. Otros protocolos como los relacionados con servicios de voz sobre IP (VoIP), marcan un paquete como perdido cuando este ha llegado con un gran retraso y ya no es necesario para reconstruir la conversación. La pérdida de paquetes es un parámetro muy importante, ya que una alta pérdida puede implicar una degradación del rendimiento debido a los mecanismos de corrección de errores implantados en protocolos de transporte tales como TCP. Por otra parte, este parámetro tiene un gran impacto en los protocolos de tiempo real, puesto que degrada la calidad de experiencia (Quality of Experience [QoE]) percibida por un usuario.

## *2.2 Medidas activas*

Las técnicas de medidas activas se basan en la idea de inyectar tráfico en la red para medir las características de la misma. Este enfoque es válido para un gran número de redes, pues se obtienen estadísticas fiables del segmento de red analizado. El principal problema de las medidas activas es que son muy intrusivas debido a su propia naturaleza. Esta característica no es deseable en algunos escenarios, ya que el comportamiento del enlace está siendo modificado por las propias medidas de tráfico. Por otro lado, en algunas ocasiones, los parámetros de calidad de servicio deben estimarse con precisión en un determinado periodo de tiempo. Por ejemplo, la evaluación de cumplimiento de acuerdos de nivel de servicio (Service-Level Agreements [SLA]) sobre un enlace específico, requiere el uso de medidas activas para obtener parámetros tales como la capacidad del enlace, retardo en un sentido (OWD) o pérdida de paquetes. Algunos de estos parámetros, como la capacidad del enlace, no pueden estimarse con medidas pasivas puesto que los enlaces no se encuentran totalmente cargados. En estos casos la estimación obtenida de los parámetros se realiza de una manera menos precisa.
En algunas situaciones la intrusión es deseable, por ejemplo, cuando se intenta estresar la red para caracterizar el mal uso de los recursos o su disponibilidad. Las medidas activas se pueden utilizar periódicamente para probar y analizar las redes. Este proceso conduce a una estratificación de las medidas activas a la hora de caracterizar el comportamiento de la red a lo largo del tiempo en función de su estado. Las técnicas de medidas activas se dividen en dos grandes grupos: las técnicas de transferencia de fichero/Bulk Data Transfer y las técnicas basadas en pares de paquetes. Esta división viene determinada por los métodos que utiliza cada grupo para generar y analizar tráfico con el objetivo de estimar los parámetros de calidad de servicio.

- **Descarga de fichero**

Este método de medida está formalmente definido por la European Telecommunications Standards Institute (ETSI) EG 202 057-4 [17]. El método de descarga de fichero tiene como objetivo estimar los parámetros de calidad de servicio utilizando una descarga HTTP. La transferencia debe hacerse consultando a un servidor de test y descargando un fichero. El fichero descargado debe ser ocho veces el ancho de banda nominal del enlace. Este fichero debe ser aleatoriamente generado para evitar que sea óptimo en cualquier servidor web. La principal ventaja de este método es que las medidas se realizan a nivel de usuario, lo cual proporciona una idea muy clara sobre la experiencia de usuario. Las técnicas de descarga de ficheros son muy fáciles de implementar, pero tienen dos inconvenientes. El primer inconveniente es que los tiempos de descarga son largos (en el orden de ocho segundos). El segundo inconveniente tiene que ver con una alta influencia del tráfico cruzado. Esto es esperable, ya que TCP realiza un ajuste del throughput basándose en el número de conexiones concurrentes.

- **Pares de paquetes**

El método de pares de paquetes es un método de medida activa basado en la idea de enviar múltiples pares de paquetes desde un origen a un destino, con el objetivo de calcular los parámetros de calidad de servicio a partir del análisis de características temporales de los paquetes. Cada par de paquetes enviado tiene el mismo tamaño y estos paquetes se envían back-to-back, es decir, a la máxima velocidad permitida. La dispersión entre cada par de paquetes (Δr) se define como el tiempo entre el último bit del primer paquete y el último bit del segundo paquete. Una ventaja de utilizar este método es que las medidas se completan en poco tiempo. Por ejemplo, las medidas realizadas en un enlace que tiene un ancho de banda de 10 Mbps duran menos de un milisegundo. La figura 2.1 muestra el comportamiento del método pares de paquetes.
Este tipo de medidas permiten la estimación de otros parámetros de calidad de servicio como OWD, variación del retardo o tasa de paquetes perdidos. Para calcular OWD se realiza la resta entre el tiempo de llegada y el tiempo de salida de cada paquete. Para estimar la variación del retardo pueden utilizarse los métodos descritos en la sección 2.1, como en el caso de OWD que se utiliza para calcular la tasa de paquetes perdidos. Esta tasa se calcula numerando secuencialmente los paquetes en el punto de medida de origen y comprobando que los paquetes llegan en el orden de numeración en el punto destino. Para calcular RTT se debe sumar el tiempo OWD de las dos direcciones de envío.
Por lo general, el método de pares de paquetes se implementa utilizando UDP como protocolo de transporte, a diferencia del método descarga de fichero, el cual utiliza TCP.
El mayor inconveniente de este método es el impacto del tráfico interferente durante el periodo de medidas.

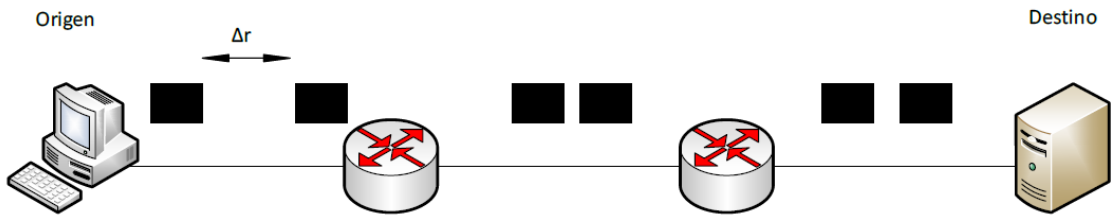



Figura 2.1 Método pares de paquetes [3].

- **Tren de paquetes**

El método descrito anteriormente es muy susceptible al tráfico cruzado, lo que motiva a usar este nuevo método tren de paquetes [18, 19, 20, 21]. Utilizando el método de pares de paquetes, hay un solo margen entre paquetes, que puede ser fácilmente rellenado por tráfico cruzado. Con el fin de disminuir las posibilidades de que el tráfico cruzado rellene el hueco entre paquetes de medida consecutivos, enviamos en su lugar un tren de N paquetes. La figura 2.2 muestra el comportamiento del método tren de paquetes. Este método constituye una técnica robusta contra el tráfico cruzado, aunque no totalmente inmune. Cuando el número de paquetes en el tren crece, la probabilidad de que cada hueco entre paquetes de medida sea ocupado con tráfico cruzado disminuye. Sin embargo, los

trenes con un gran número de paquetes tienen un efecto negativo como se muestra en [22]. Los trenes de paquetes de gran longitud son excesivamente intrusivos. Por lo tanto, debe haber un compromiso entre intrusión e inmunidad al tráfico cruzado.

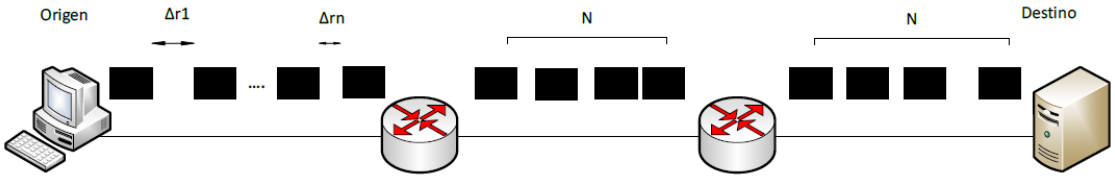



Figura 2.2 Método tren de paquetes [3].

## *2.3 Medidas pasivas*

Las medidas pasivas se basan en la recolección de tráfico de la red para su posterior análisis en términos de rendimiento y comportamiento. La principal ventaja de este método es que se aborda la tarea de monitorización de red de forma no intrusiva. La estimación precisa de parámetros de calidad de servicio utilizando medidas pasivas presenta un nivel alto de dificultad, debido a la variación de las condiciones en función del intervalo temporal utilizado para el análisis. Los datos recogidos pueden pertenecer a cualquiera de las siguientes categorías:

- Tráfico capturado directamente: por ejemplo, trazas Packet Capture (PCAP) o paquetes capturados utilizando hardware especializado como FPGAs o Network Processors.

- Datos y estadísticas pre-procesadas procedentes de dispositivos: esta información puede ser obtenida de routers, switches o sondas instaladas en diversos puntos de la red. Por ejemplo, datos similares a los recolectados por la herramienta Multi Router Traffic Grapher (MRTG) o flujos como los generados por sistemas como NetFlow de Cisco.

Dependiendo del tráfico recogido, la monitorización pasiva puede dar lugar a análisis con diferentes niveles de profundidad. Por ejemplo, analizando la información proveniente de un sistema de detección de intrusos en una red (Network Intrusion Detection System [NIDS]), se pueden obtener como datos de salida: el número de amenazas detectadas, una lista de direcciones IP maliciosas activas o el número de conexiones del protocolo de transferencia de hipertexto (Hyper Text Transfer Protocol [HTTP]) sospechosas por segundo. Por otro lado, analizando datos de bajo nivel, las estimaciones pueden dar una salida ligeramente diferente. En este escenario, por ejemplo, algunas estimaciones de parámetros de calidad de servicio como el número de paquetes perdidos o el ancho de banda agregado pueden ser obtenidos, así como otras estimaciones como el número de flujos activos o una lista con las direcciones IP más activas. El análisis paquete a paquete de todo el tráfico que atraviesa los enlaces monitorizados puede proveer toda la información anterior. Normalmente este enfoque es muy costoso tanto en términos de almacenamiento como de potencia computacional requerida para capturar y analizar en tiempo real.
La mayor ventaja de las técnicas pasivas es que no son intrusivas. Usando el analizador de puertos del switch (Switched Port Analyzer [SPAN]), el tráfico en los routers y switches

puede ser capturado sin interferencias. Sin embargo, en algunos casos, una cantidad de tráfico extra debe ser introducida en la red para realizar el transporte y la recolección de datos como los que obtiene MRTG. Este tráfico añadido se considera menos intrusivo que las técnicas de medida activas.

- **Monitorización a nivel de flujos**

El proceso de monitorización a nivel de flujo se basa en los protocolos NetFlow o IPFIX para exportar información de routers o switches y poder estimar así cada uno de sus parámetros de calidad de servicio o hacer hipótesis sobre el estado y rendimiento de la red monitorizada. Utilizando esta información se han propuesto algunos enfoques para estimar los parámetros de calidad de servicio.

Para la recolección y representación de información de medidas de flujos existen tres herramientas destacables en el estado del arte: Flow-tools, FlowScan, Cflowd.

Flow-tools es una colección de programas que se utilizan para crear procesos compatibles con Cisco Netflow. Dentro de este conjunto de programas podemos encontrar programas tales como flow-capture que se encarga de recoger los datos exportados de NetFlow y almacenarlos en disco, además de gestionar el espacio en el mismo. Otro programa que podemos encontrar es flow-fanout. Este programa se encarga de replicar NetFlow UDP flujos de una fuente a muchos destinos, el destino puede ser una dirección de multidifusión. Flow-expire, elimina los flujos más antiguos basándose en el uso del disco. A la hora de visualizar los datos, Flow-tools cuenta con la herramienta flow-print, la cual genera ficheros de flujos con el formato que se muestra en la figura 2.3 [23]. También puede realizar filtrados por características a nivel de flujo con la herramienta flow-filter.

```
eng1:% flow-print < ft-v05.2002-01-21.093345-0500 | head -15
srcIP            dstIP            prot  srcPort  dstPort  octets packets
131.238.205.199  194.210.13.1     6     6346     40355    221          5
192.5.110.20     128.195.186.5    17    57040    33468    40           1
128.146.1.7      194.85.127.69    17    53       53       64           1
193.170.62.114   132.235.156.242  6     1453     1214     192          4
134.243.5.160    192.129.25.10    6     80       3360     654          7
132.235.156.242  193.170.62.114   6     1214     1453     160          4
130.206.43.51    130.101.99.107   6     3226     80       96           2
206.244.141.3    128.163.62.17    6     35593    80       739          10
206.244.141.3    128.163.62.17    6     35594    80       577          6
212.33.84.160    132.235.152.47   6     1447     1214     192          4
132.235.157.187  164.58.150.166   6     1214     56938    81           2
129.1.246.97     152.94.20.214    6     4541     6346     912          10
132.235.152.47   212.33.84.160    6     1214     1447     160          4
130.237.131.52   130.101.9.20     6     1246     80       902          15
```

Figura 2.3: Visualización de datos Flow-tools.

Cflowd es una herramienta de análisis de flujos que se utiliza actualmente para el análisis de métodos de conmutación Netflow de Cisco. Esta herramienta es usada para que los proveedores de servicios de Internet (Internet Service Provider [ISP]) puedan recoger datos para la planificación de la capacidad y actividades similares, y para involucrar a los ISP en el desarrollo de herramientas más avanzadas para la representación gráfica. Cflow es una herramienta importante para la planificación de la capacidad y dimensionado de las redes

de los ISP, análisis de tendencias y caracterización de carga de trabajo, proporcionando un medio para analizar los datos de tráfico por flujo. Cflowd no tiene la suficiente granularidad para mostrar IPs de origen. Esto quiere decir que necesita usar otros programas para saber si un equipo está roto o infectado. Para la visualización de los datos recogidos por Cflow se necesita de programas tales como FlowScan, mencionado anteriormente.

## *2.4 Herramientas de monitorización*

### 2.4.1 PING

Ping, es una herramienta de diagnóstico que permite comprobar el estado de conexión con al menos un host remoto que se encuentre dentro de una red de tipo IP. Se podría decir que la utilidad de dicho comando es comprobar si una dirección IP determinada es accesible desde la red o no. El funcionamiento básico de este comando es a través del envío de mensajes ICMP contenidos en paquetes IP. Cada mensaje ICMP es identificado de forma unívoca, pudiendo así detectar los mensajes retornados. El protocolo IMCP no es un protocolo de transporte y no utiliza ningún protocolo de la capa de aplicación. Como se puede observar en la figura 2.4 a través de los tiempos de respuesta podemos obtener una estimación del parámetro de calidad de servicio RTT, lo que nos permite tener una primera impresión acerca de la calidad de comunicación con el host destino.

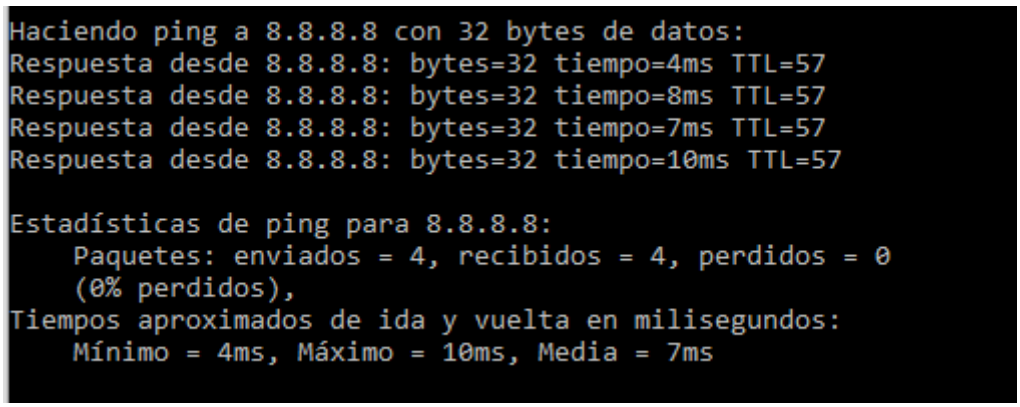



Figura 2.4: Ejemplo ping.

### 2.4.2 IPERF

Iperf es una herramienta cuya finalidad es medir el ancho de banda con TCP o UDP como protocolos de conexión. Iperf proporciona la posibilidad al usuario de medir el rendimiento entre equipos de una manera unidireccional o bidireccional. Iperf trabaja en modo cliente-servidor, esto quiere decir que ha de haber dos instancias del programa, una como cliente y otra como servidor. El servidor IPERF se queda a la escucha de peticiones del cliente. El cliente se conecta a un puerto del servidor y envía datos por el canal de subida [24]. A continuación, en las figuras 2.5 y 2.6 se pueden ver las respuestas por parte del servidor y del cliente.

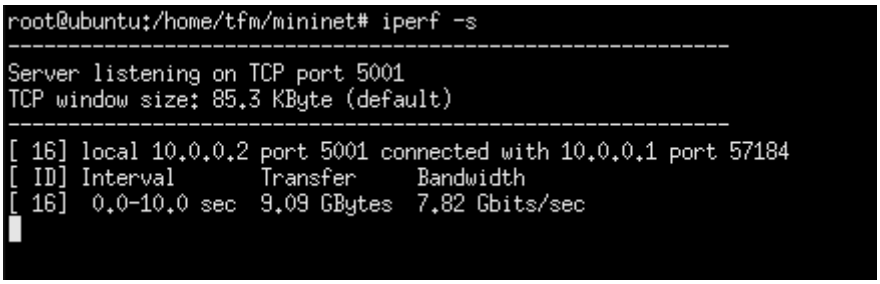



Figura 2.5: Servidor iperf.

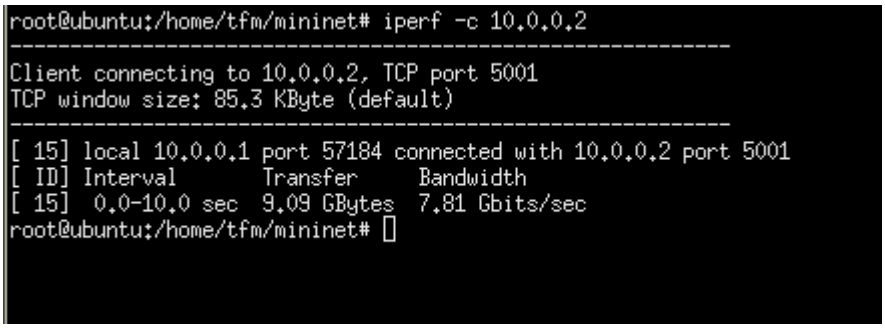



Figura 2.6: Cliente iperf.

### 2.4.3 FPROBE

FPROBE es una herramienta que permite recolectar tráfico de la red, generar flujos NetFlow, y exportarlos hacia un colector que se especifique.

NetFlow es un protocolo de red desarrollado por Cisco Systems para recolectar información sobre tráfico IP [25]. Hoy en día NetFlow es soportado por la mayoría de routers y switches Cisco. Los dispositivos hardware con la opción NetFlow habilitada se encargarán de enviar flujos NetFlow a un colector que a su vez puede estar a la escucha de otros equipos. En el caso de no disponer de un dispositivo hardware capaz de generar datos NetFlow se usan soluciones software como FPROBE. En la figura 2.7 que se presenta a continuación se puede ver la salida de dos flujos NetFlow donde el tráfico ha sido generado con la herramienta FPROBE.

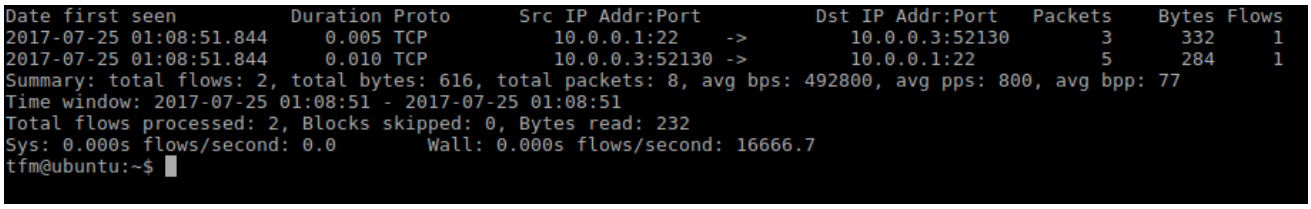



Figura 2.7: Salida pantalla FPROBE.

La herramienta se compone de 3 utilidades:

1. Fprobe consiste en un analizador de tráfico que además generará flujos Netflow y exportará mensajes NetFlow.
2. El demonio "nfcapd" recibe los mensajes Netflow y guarda la información en un formato determinado. Con esta utilidad también podremos configurar dónde guardar los archivos de flujos generados o seleccionar cada cuánto tiempo queremos volcar la información a un fichero.
3. Nfdump nos permite visualizar las medidas recolectadas por el demonio nfcapd. Esta utilidad nos permite visualizar los datos como se mostraban en la figura 2.7.

## *2.5 NETCONF y YANG*

### 2.5.1 NETCONF

El protocolo SNMP ha sido ampliamente utilizado durante décadas. Este protocolo se basa en un modelo petición-respuesta que utiliza como protocolo de transporte no orientado a la conexión, normalmente UDP. Actualmente la mayoría de redes de comunicación son de carácter complejo y de gran extensión incluyendo múltiples proveedores y distintas tecnologías de equipos. Esto afecta a SNMP pues este protocolo ya no es escalable como un método eficiente y efectivo para realizar numerosas operaciones de gestión de red. Como alternativa en los últimos años ha surgido NETCONF [26].

El protocolo NETCONF define un mecanismo simple a través del cual podemos gestionar un dispositivo de red, obtener información sobre la configuración del dispositivo y manipular o actualizar ficheros de configuración [2]. Se basa en el uso de XML utilizando un protocolo de transporte orientado a la conexión (en la mayoría de los casos TCP). Además, NETCONF puede utilizarse de manera segura a través de túneles SSH.

NETCONF utiliza un mecanismo simple de comunicación basado en RPCs entre un cliente y un servidor. El cliente codifica cada llamada RPC en un mensaje XML y se lo envía al servidor utilizando un método de transporte (generalmente seguro). El servidor responde con una réplica codificada también en XML. En la figura 2.8 se puede ver un ejemplo de mensaje RPC en el que se utiliza el método "get" con un atributo adicional llamado "user_id". Si nos fijamos en el mensaje de respuesta vemos que se devuelve el atributo "user_id" como el contenido solicitado. El cliente puede ser un script o una aplicación típica corriendo dentro de un software gestor de red. El servidor normalmente suele estar localizado en un dispositivo de red. Se denomina sesión NETCONF a la conexión lógica entre un cliente y un servidor. Cada servidor soporta al menos una sesión. Una vez se establece la sesión se podrán cambiar los parámetros de configuración que se deseen.

```
<rpc message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:ex="http://example.net/content/1.0"
     ex:user-id="fred">
  <get/>
</rpc>

<rpc-reply message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
     xmlns:ex="http://example.net/content/1.0"
     ex:user-id="fred">
  <data>
    <!-- contents here... -->
  </data>
</rpc-reply>
```

Figura 2.8: Mensajes NETCONF [2].

Para el funcionamiento del protocolo se define el siguiente modelo basado en capas [2] (el diagrama se muestra en la Figura 2.9):

- La capa de transporte proporciona una ruta de comunicación entre el cliente y el servidor. Cabe destacar en este punto una cualidad importante de NETCONF pues nos permite utilizar la mayoría de protocolos de transporte.

- La capa de mensajes proporciona un mecanismo de trazas de transporte independiente de la codificación de los mensajes RPC y las notificaciones. Dentro del elemento <rpc> se encuentra una solicitud NETCONF que se envía desde el cliente hacia el servidor. El servidor debe responder al mensaje del cliente con el elemento <rpc-reply>. Aparte de las RPC explícitas, también existen notificaciones que se envían como respuestas a eventos.

- La capa de operación define un conjunto de procedimientos invocados como métodos RPC con parámetros de codificación XML. En otras palabras, NETCONF proporciona un conjunto pequeño de operaciones para manejar los dispositivos. Estas tareas vienen definidas en la capa de operación, algunas de estas operaciones son: get, get-config, delete-config, lock, close-session.

- La capa de contenido y parte de la capa de operación están cubiertas por el lenguaje de modelo de datos YANG.

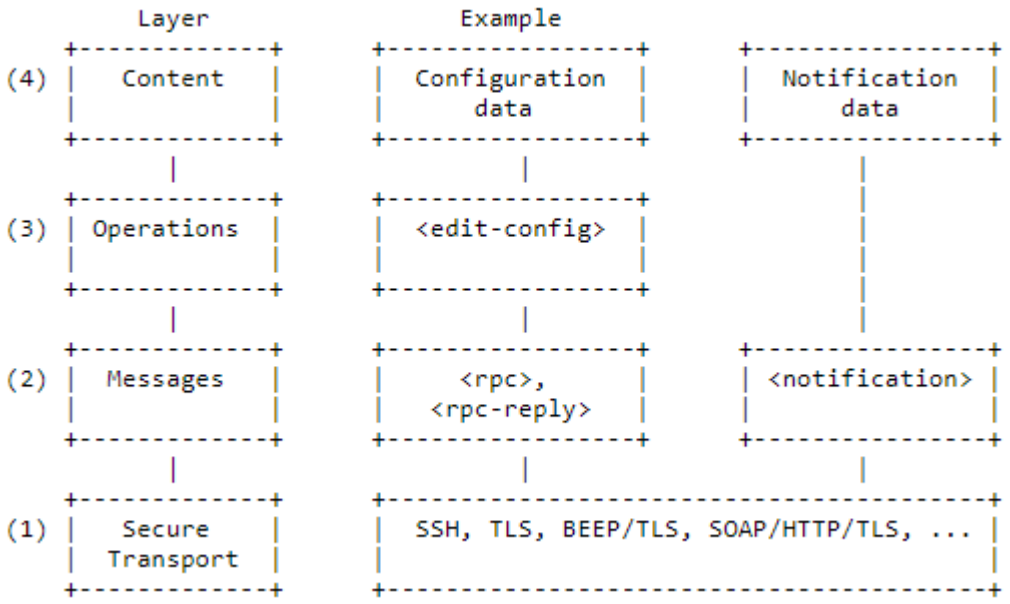



Figura 2.9: Capas NETCONF [2].

### 2.5.2 YANG

YANG [27] es un lenguaje de modelado de datos usado junto con el protocolo NETCONF. Un modelo de datos YANG define una jerarquía sobre los datos que pueden ser usados para operaciones base: incluir configuración, estado de datos, llamadas a procedimientos remotos (RPC) y notificaciones. Esto permite una completa descripción sobre todos los datos que son enviados entre un cliente y un servidor NETCONF.

El modelo de datos YANG organiza los datos en forma de árbol en el que cada nodo tiene un nombre y un valor o un conjunto de nodos hijos. YANG proporciona una clara descripción de los nodos y define como se interactúa entre ellos.

YANG estructura los modelos en módulos y submódulos. Un módulo puede importar datos de otro módulo externo e incluir datos de submódulos. La jerarquía puede aumentarse, permitiendo que un módulo agregue nodos de datos a la jerarquía definida en otro módulo. Este aumento puede ser condicional con nuevos nodos que aparecen solo si se cumplen ciertas condiciones.

Los módulos YANG pueden ser traducidos en su equivalente en XML. Esta sintaxis tiene el nombre de YANG Independent Notificacion (YIN), permitiendo a las aplicaciones utilizar herramientas que actúen sobre datos en XML que son más comunes. La conversión entre el modelo de datos YING y YANG no tiene pérdidas, por lo que el modelo convertido a YIN puede volver a transformase en YANG.

```
<rpc name="iperf_s">
  <description>
    <text>Arranca un servidor de iperf</text>
  </description>
</rpc>
<rpc name="iperf_c">
  <description>
    <text>Arranca cliente iperf</text>
  </description>
</rpc>
<rpc name="iperf_stop">
  <description>
    <text>Detiene servidores iperf</text>
  </description>
</rpc>
```

Figura 2.10: Ejemplo YIN.

```
rpc iperf_s {
  description
    "Arranca un servidor de iperf";
}
rpc iperf_c {
  description
    "Arranca cliente iperf";
}
rpc iperf_stop {
  description
    "Detiene servidores iperf";
}
```

Figura 2.11: Ejemplo YANG.

### 2.5.3 Implementación del protocolo NETCONF

Existen pocas implementaciones libres del protocolo NETCONF. En este proyecto se hará uso de la librería Libnetconf. Libnetconf es una librería que implementa el protocolo NETCONF desarrollada en C. La librería está diseñada para construir servidores y clientes NETCONF. Su API proporciona funciones básicas para conectar un cliente y un servidor NETCONF utilizando SSH o TLS y poder enviar y recibir mensajes NETCONF. Respecto a la comunicación mediante SSH la librería permite el uso nativo de funciones de la librería libssh o la creación de subsistemas SSH standalone compatibles con OpenSSH. Por motivos de flexibilidad y simplicidad, ésta es la opción usada en este trabajo.

La librería Libnetconf provee un framework adicional llamado transAPI que está diseñado para ayudar a los programadores permitiéndoles de una manera sencilla configurar y manejar dispositivos sin tener que enfrentarse directamente al protocolo NETCONF. Este framework permite al desarrollador crear bloques funcionales de código que se integren de manera automática dentro de la librería Libnetconf. Basándose en una lista llamada “rutas

sensibles" el generador crea un fichero en código C que contiene un *stub* con funciones de callback para cada "ruta sensible" definida. Cada vez que algo cambia en el archivo de configuración, se realiza una llamada a la función específica de callback, lo que producirá cambios en la configuración dando lugar a cambios en el comportamiento del dispositivo. Este mecanismo se utiliza generalmente para implementar notificaciones.

Adicionalmente el framework transAPI proporciona una oportunidad de implementar operaciones NETCONF RPC definidas a partir de un modelo de datos. En este caso haciendo uso de la herramienta lnctool proporcionada junto con la librería. Se buscan definiciones de RPCs dentro de un modelo de datos dado y se generan *stubs* con funciones de callback para cada RPC definido en el modelo. Dichas funciones de callback se compilan y enlazan en una librería dinámica que se integra con Libnetconf mediante un sistema de registro de RPCs. Cuando un servidor recibe una petición RPC, se busca si existe una RPC registrada y, en caso de que exista, se llama a la función de callback correspondiente [28]. La encargada de gestionar los datos de entrada y salida de la RPC es la función de callback y queda definida por la implementación que se lleve a cabo. Este flujo queda esquematizado en la figura 2.12.

Esta última funcionalidad es la que se ha utilizado en este proyecto a la hora de manejar y configurar sondas de red.

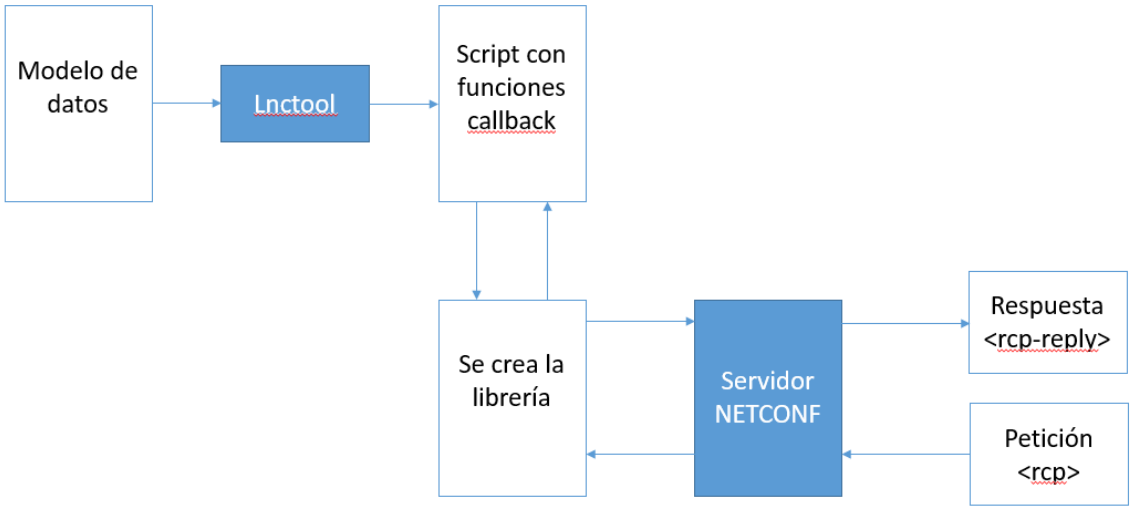



Figura 2.12: Flujo transAPI, libnetconf.

## *2.6 Antecedentes*

Existen varias aproximaciones en la literatura sobre como configurar y monitorizar sondas utilizando NETCONF y YANG. El proyecto [29] viene motivado por el hecho de haber en la actualidad muchas aplicaciones que necesitan beneficiarse de poder adaptar sus ficheros de configuración a la demanda del mercado. Para ello han utilizado el protocolo NETCONF y creado un fichero XML para cada dispositivo del que pueden modificar sus parámetros.

Otro proyecto [30] demuestra la posibilidad de gestionar sondas usando el protocolo NETCONF. SamKnows es un elemento de una plataforma de medición que realiza medidas activas utilizando un hardware propio para evaluar el ancho de banda de la red. En la actualidad existen alrededor de 20000 sondas por todo el mundo con este hardware.
Dentro del grupo de trabajo que presenta este proyecto existió una fuerte inclinación por el uso de protocolos existentes para manejar estas sondas. Se consideró utilizar NETCONF para la gestión de las mismas, sin embargo, no todos los routers domésticos soportan NETCONF. Para resolver este problema se desarrolló el servidor NETCONF utilizando la librería libnetconf acondicionada al hardware de las sondas que manejaban. A continuación, se instaló NETCONF en las sondas de medición demostrando así, como un cliente puede usarse para configurar dichas sondas. Para una configuración básica de las sondas se ha utilizado el modelo de datos YANG. Este modelo de datos especifica la configuración y el estado de los datos utilizados para establecer y recuperar información sobre el sistema. Además, el modelo define nuevas operaciones NETCONF para apagar o iniciar el dispositivo.

Estos proyectos utilizan NETCONF para la comunicación con sondas de medidas de red. El trabajo que aquí se detalla se diferencia del resto en la forma de obtener la visión global del estado de la red, pues se utilizan específicamente tres herramientas de gestión de medidas con la que se ha considerado que se obtiene un informe completo de la red. Principalmente esta propiedad viene determinada por el uso de RPCs. Esto da la posibilidad de ejecutar cualquier herramienta de red estándar. En el proyecto desarrollado se trabaja también con una gran variedad de métodos con los que se podrán obtener parámetros de calidad de servicio. Esto resulta característico pues la mayoría de proyectos solo hablan de técnicas de medidas activas y pasivas sin llegar al detalle de los métodos.

Por otro lado, se ofrece al usuario una interfaz web de fácil manejo con la que podrá gestionar sondas en tiempo real y obtener históricos de medidas.

# 3 Diseño

En esta sección entraremos a describir el diseño llevado a cabo para cumplir los objetivos propuestos: creación de modelos de datos que representen operaciones, configuraciones y datos de medida que pueden ser obtenidos en las sondas de monitorización Ethernet. Además, se mostrarán los casos de uso del entorno web.

## *3.1 Sistema web*

Para desarrollar el entorno web se propone el uso de Django. Se trata de un framework web de código abierto escrito en Python, que permite construir aplicaciones web de manera rápida, limpia y estructurada. El paradigma de Django se basa en automatizar todo lo posible tareas comunes y se adhiere al principio DRY (Don´t Repeat Yourself). La idea principal del entorno web es proporcionar una interfaz que permita realizar acciones que después se traduzcan en operaciones a bajo nivel que a su vez se encapsulen dentro de mensajes NETCONF.

En la figura 3.1 se puede visualizar el flujo de trabajo que realiza la aplicación cuando un usuario ejecuta una herramienta de medida (por ejemplo, ping). A continuación, se define el proceso de manera desglosada;

1. Cuando un usuario quiere ejecutar una herramienta de medida desde el entorno web a través de un navegador, éste manda una solicitud.
2. La URL utilizada es gestionada por Django. Django se encarga de asociar cada URL con una vista. Una vista Django es una función Python que toma una petición web, la procesa y devuelve una respuesta web.

Las vistas, generalmente, interactúan con un modelo para obtener datos y poder generar la respuesta. Un modelo se define como el conjunto de datos y comportamientos que componen la información que maneja el sistema. Generalmente cada modelo se mapea a una tabla de una base de datos.

3. Las vistas ejecutarán un programa cliente que hará uso de libnetconf para enviar una petición RPC contra el servidor NETCONF localizado en la sonda de monitorización.
4. El servidor validará la petición, ejecutará la herramienta y devolverá mediante un mensaje NETCONF la respuesta.
5. La respuesta es retornada a la vista, la cual podrá interactuar con los datos devueltos para realizar una disgregación de la misma y almacenar los valores de interés en una base de datos.
6. Una vez obtenidos los datos, la vista puede hacer uso de una plantilla que permita construir la respuesta web de acuerdo con una estructura y recursos concretos.
7. La respuesta se envía al navegador y este la muestra del modo que corresponda.

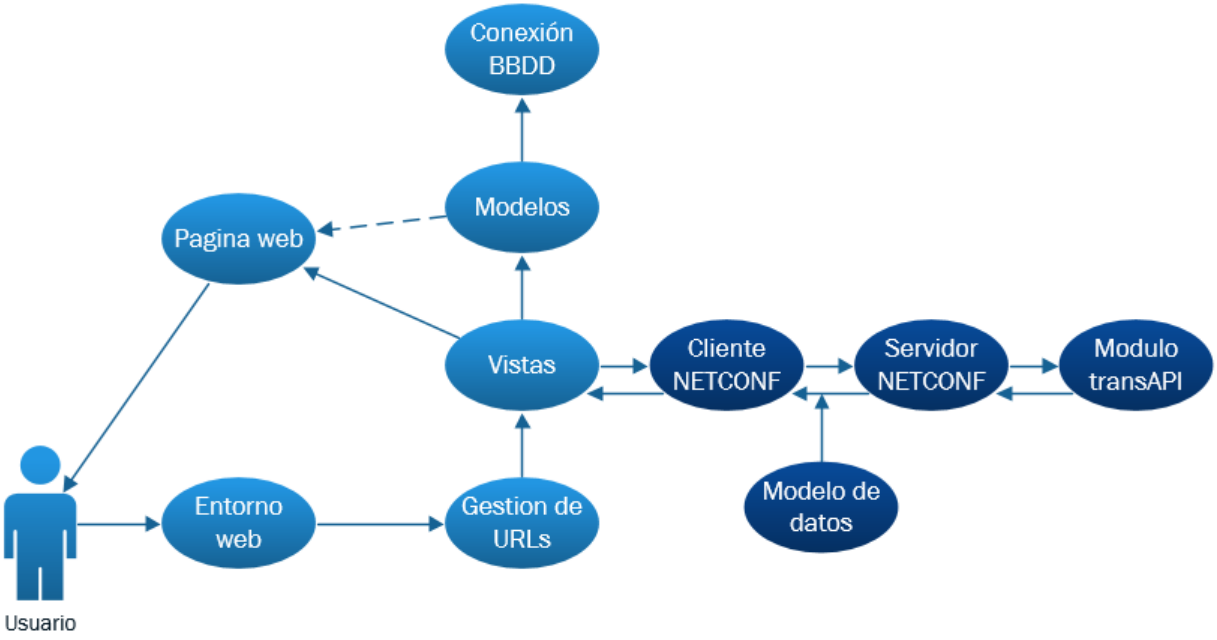



Figura 3.1: Flujo aplicación.

A continuación, en el apartado de desarrollo veremos con más detalle cómo interactúa el cliente con el servidor NETCONF y cuál es la funcionalidad del módulo transAPI.

El diseño que se eligió para las diferentes páginas web se puede visualizar sintetizado en la figura 3.2. Para todas las páginas web se dará una breve descripción del funcionamiento de la herramienta de análisis que se esté utilizando. Seguidamente se muestran los botones de acción donde el usuario podrá tomar las medidas de red, consultar el histórico, visualizar gráficas y realizar más acciones según la página que consulte. Para mostrar las gráficas se propone la utilización de HighCharts [31]. Highcharts es una librería Javascript que permite representar datos gráficamente, hacer zoom en una sección de la gráfica, así como imprimirla o guardarla. La librería es de código abierto y de uso libre siempre que no se trabaje con ella para fines comerciales.

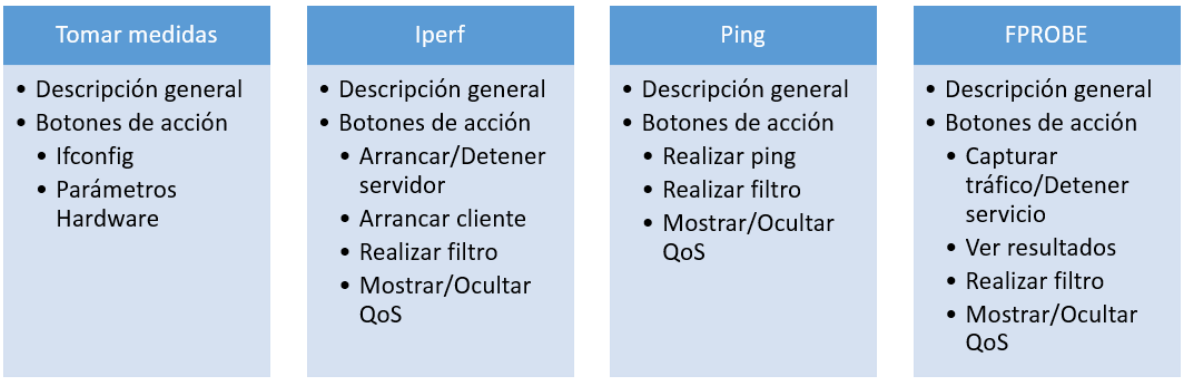



Figura 3.2: Diseño páginas web.

La interfaz web permite que la herramienta desarrollada en este proyecto sea accesible desde cualquier lado y utilizando cualquier sistema operativo. También facilita la integración de manera sencilla con múltiples sistemas de visualización, además de permitir la creación de aplicaciones cliente para sistemas operativos móviles.

## *3.2 Aplicaciones NETCONF*

Viendo el proyecto desde una perspectiva general y centrándonos en el uso del protocolo NETCONF, el diseño que se plantea es el mostrado en la figura 3.3, donde el flujo es el siguiente: desde el entorno web o front-end se ejecuta una herramienta de análisis que levanta un cliente NETCONF. El cliente se conectará con la sonda desplegada que se desee y ejecutará la operación que se haya seleccionado. El resultado será devuelto a la vista de Django donde se insertará en la base de datos para devolver el resultado al usuario.

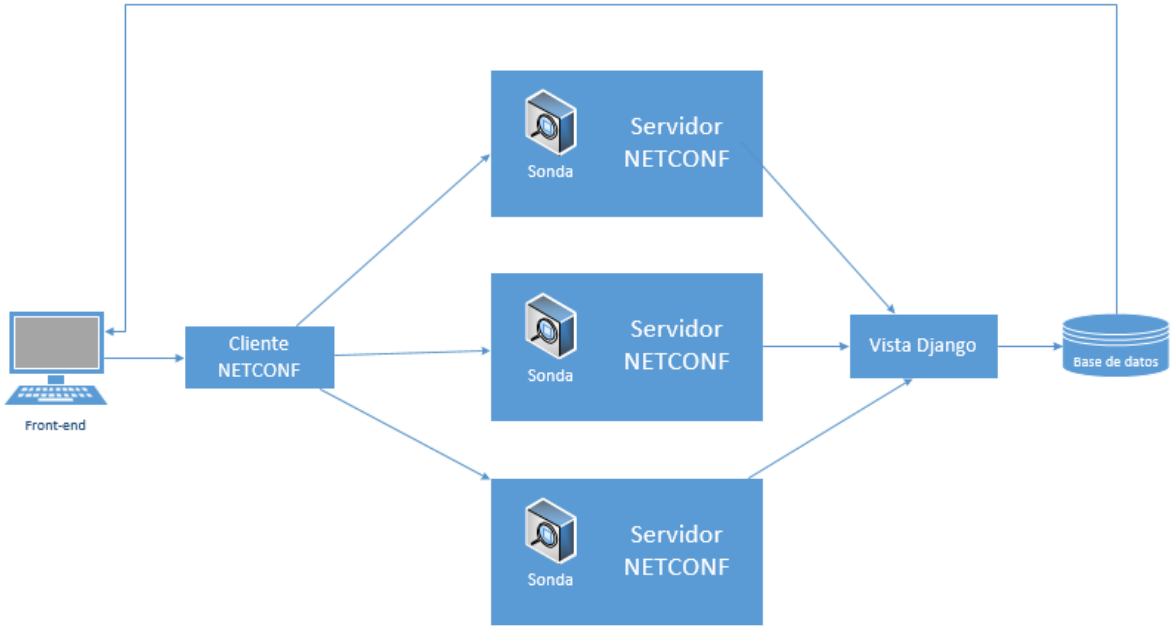



Figura 3.3: Diagrama general.

Para las tres posibles herramientas de análisis (PING, IPERF y FPROBE) usadas en este trabajo se han tenido en consideración los siguientes parámetros y datos:

| | PING | IPERF | FPROBE |
|---|---|---|---|
| IP origen | ✓ | ✓ | ✓ |
| IP destino | ✓ | ✓ | ✓ |
| Puerto origen | | ✓ | ✓ |
| Puerto destino | | ✓ | ✓ |
| Perdida de paquetes | ✓ | | |
| RTT mínimo | ✓ | | |
| RTT medio | ✓ | | |
| RTT máximo | ✓ | | |
| RTT desviación | ✓ | | |
| Timestamp | ✓ | ✓ | |
| Duración | | ✓ | |
| Ancho de banda | | ✓ | |
| Protocolo transporte | | | ✓ |
| Contador de bytes | | | ✓ |
| Contador de paquetes | | | ✓ |
| Tiempo inicio | | | ✓ |
| Tiempo fin | | | ✓ |

Tabla 3.1: Parámetros de calidad de servicio.

Con estas tres herramientas podemos conocer parámetros de calidad que nos dan una visión amplia del estado de la red. Ping e Iperf son técnicas de medidas activas de las que podemos obtener parámetros como el RRT y el ancho de banda. Por el contrario, con FPROBE podemos capturar los flujos que atraviesan un nodo de la red, siendo esto una técnica de medida pasiva.

Para definir las operaciones NETCONF que se pueden ejecutar desde el entorno web se ha desarrollado un modelo de datos YANG. Se puede consultar el modelo de datos en el Anexo I.

## *3.3 Casos de uso*

A continuación, se describirán los casos de uso más relevantes para el sistema diseñado. Cuando un usuario acceda al entorno web va a poder realizar medidas en tiempo real utilizando las herramientas de análisis de red PING, IPERF y FPROBE. Cuando ejecute cada una de estas operaciones podrá visualizar el resultado al instante. También se dará la posibilidad al usuario de consultar el histórico y obtener gráficas sobre el mismo. Si se desea se puede obtener un histórico y una gráfica filtrada por algunos de los campos de interés. Además, el entorno web permite consultar históricos utilizando diferentes técnicas de análisis de red. A continuación, en la figura 3.4 a través del diagrama de caso de uso, se pueden resumir todas las funcionalidades que se le dan al usuario cuando accede al entorno web.

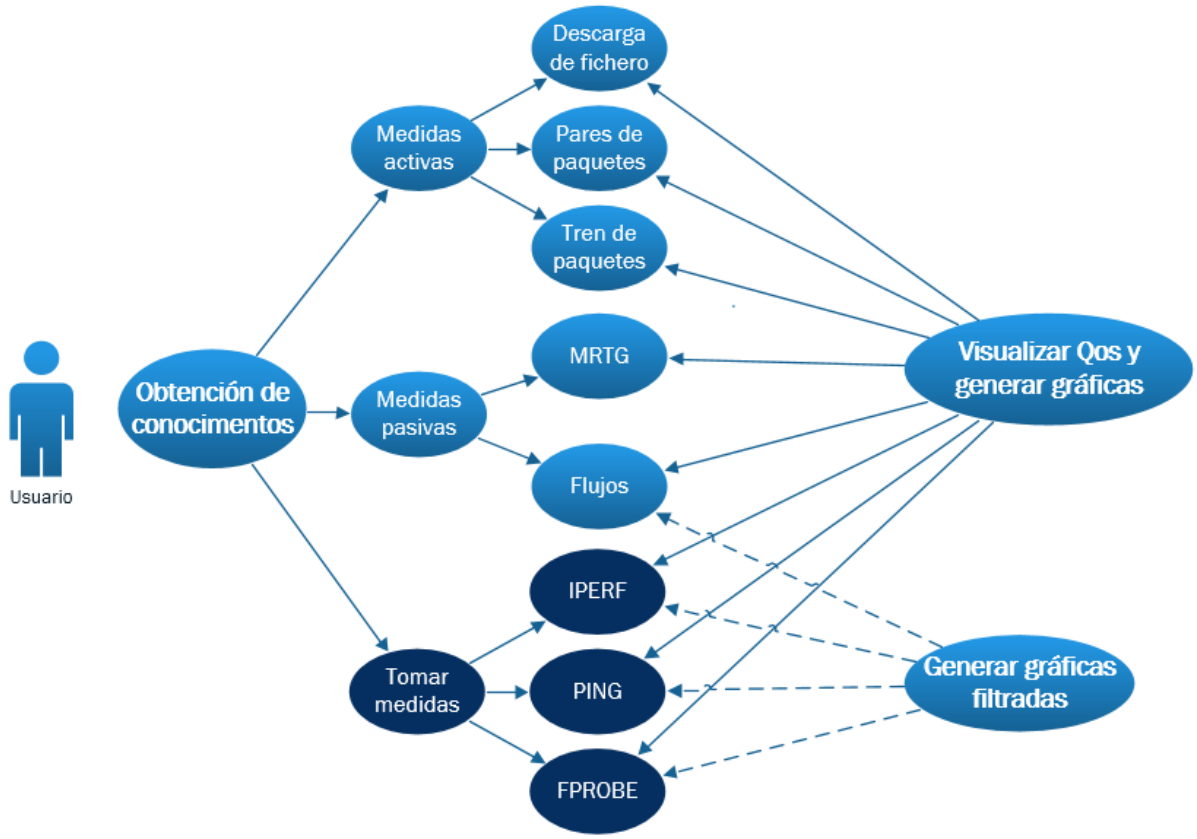



Figura 3.4: Caso de uso.

# 4 Desarrollo

En esta sección se detallará en más profundidad el desarrollo de este proyecto. Veremos cómo es el front-end desde un punto de vista de usuario. Se explicará de manera resumida el framework transAPI y por último, se hablará sobre el entorno configurado para pruebas.

## *4.1 Desarrollo del Front-end*

Para este proyecto se ha tomado como base el entorno web desarrollado en [3]. Este entorno utiliza una base de datos SQLite, ya que en el proyecto anteriormente mencionado se demostró que se obtiene un mejor rendimiento que con otras bases de datos.

Sobre el menú inicial se ha añadido la opción "*Tomar medidas*"


Figura 4.1: Inicio entorno web.

Esta nueva sección incluye cuatro páginas: "*Tomar medidas*", "*PING*", "*IPERF*" y "*FPROBE*". En la primera página se incluye una descripción general, además se da la

posibilidad al usuario de ejecutar el comando UNIX “IFCONFIG” y obtener, además, características del hardware sobre la sonda que se desee.


Figura 4.2: Tomar medidas.

Si se selecciona el botón “*Realizar ifconfig*” aparecerá un formulario que nos permitirá introducir una IP de las sondas desplegadas. Cuando se pulsa en “*Ejecutar*” se recibe información sobre la interfaz red y su dirección MAC como se puede ver en la figura 4.3.


Figura 4.3: IFCONFIG.

Si se selecciona el botón "*Obtener propiedades*" obtendremos el mismo formulario, pero esta vez, el resultado será información sobre el hardware de la sonda tal y como se muestra en la figura 4.4.


Figura 4.4: Propiedades sonda.

En la pantalla de "*Iperf*" podemos ver una descripción de la herramienta de análisis seleccionada, así como cuatro botones de acción con diferentes funcionalidades: "*Arrancar/Detener servidor*", "*Arrancar cliente*", "*Realizar filtro*" y "*Mostrar/Ocultar QoS*".


Figura 4.5: Iperf.

El primer botón, como su nombre indica, nos permitirá arrancar un servidor de IPERF. Para ello se deberá introducir una IP y pulsar en "*Ejecutar*". También nos permite detener los servidores que haya en ese momento en ejecución, pulsando sobre el botón "*Detener servicio*". Ver figura 4.6, resultado de ejecutar un servidor y figura 4.7 para ver el resultado del servidor detenido.


Figura 4.6: Arrancar servidor iperf.


Figura 4.7: Detener servidor iperf.

Cuando seleccionamos "*Arrancar cliente*" en el formulario nos pedirá que introduzcamos la dirección IP del servidor previamente levantado y la IP donde se quiera invocar al cliente. Una vez seleccionemos "*Medir*" en el entorno web se mostrarán en una tabla los resultados de la medición como se puede observar en la figura 4.8.

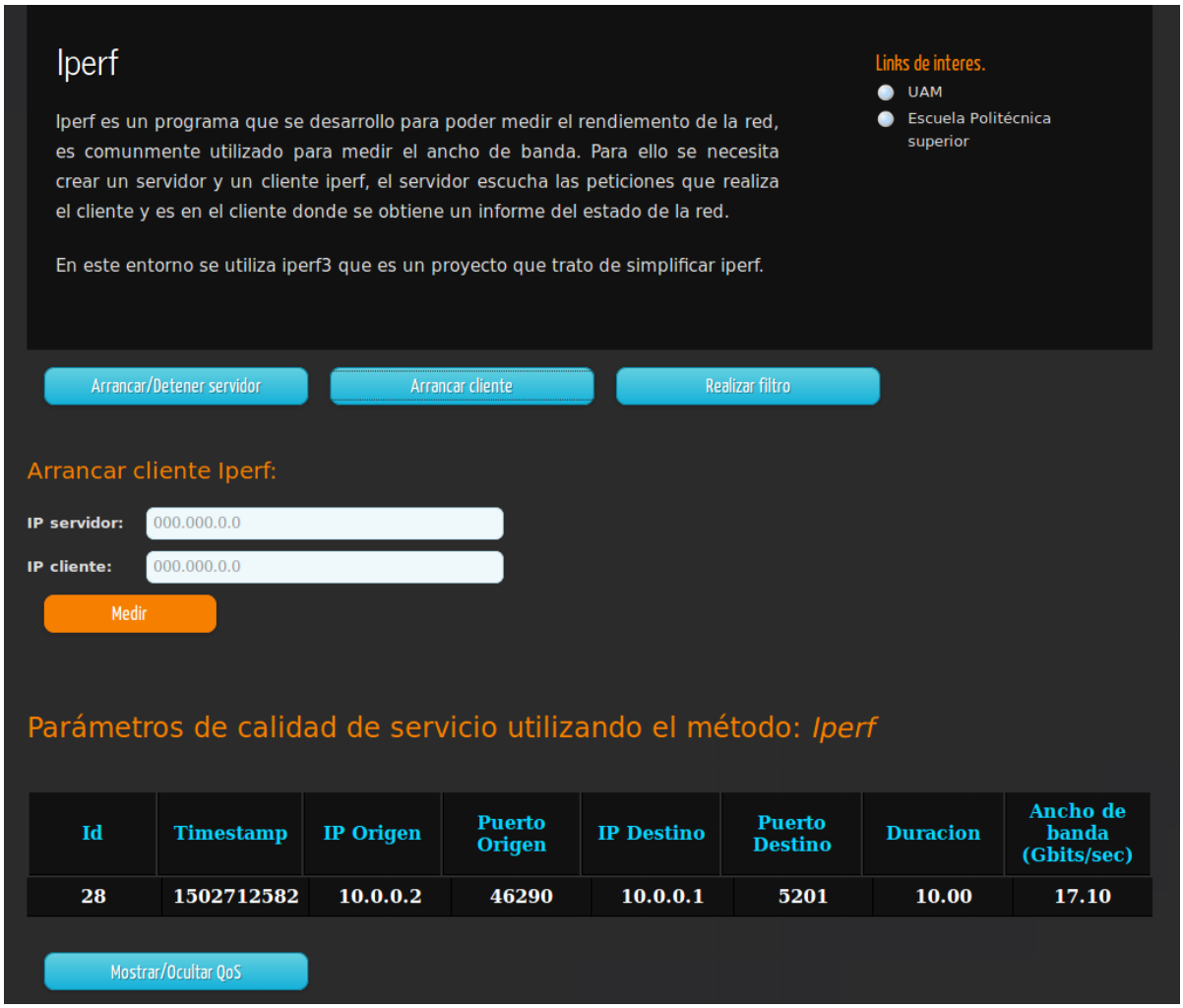



Figura 4.8: Resultado iperf.

Este resultado se añadirá al histórico para su posterior estudio.


Figura 4.9: Histórico iperf.

El botón "*Realizar filtro*" da la posibilidad de realizar filtros sobre el total del histórico por los campos disponibles en el formulario: "*IP origen*", "*IP destino*", "*Puerto origen*" y "*Puerto destino*".

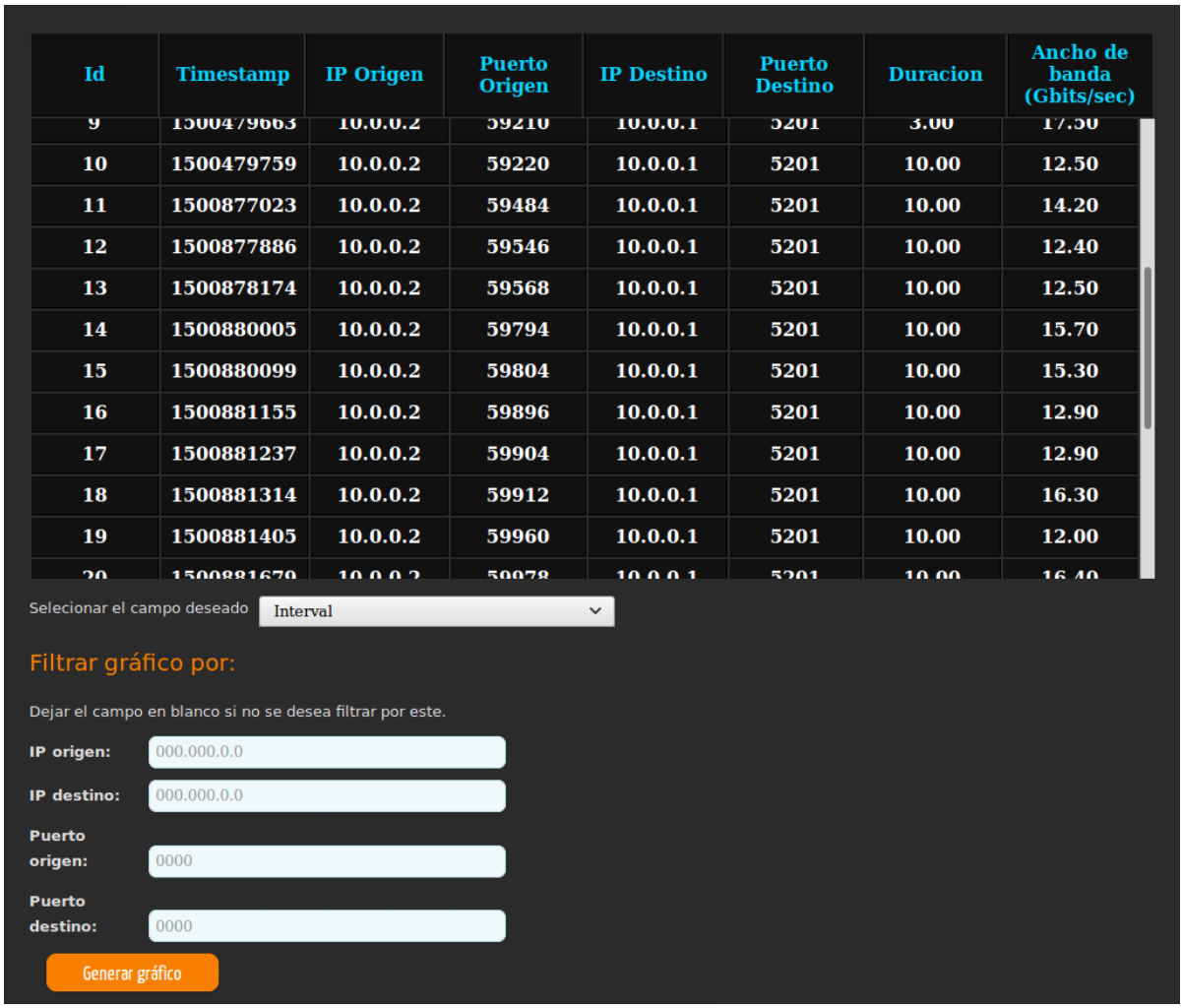
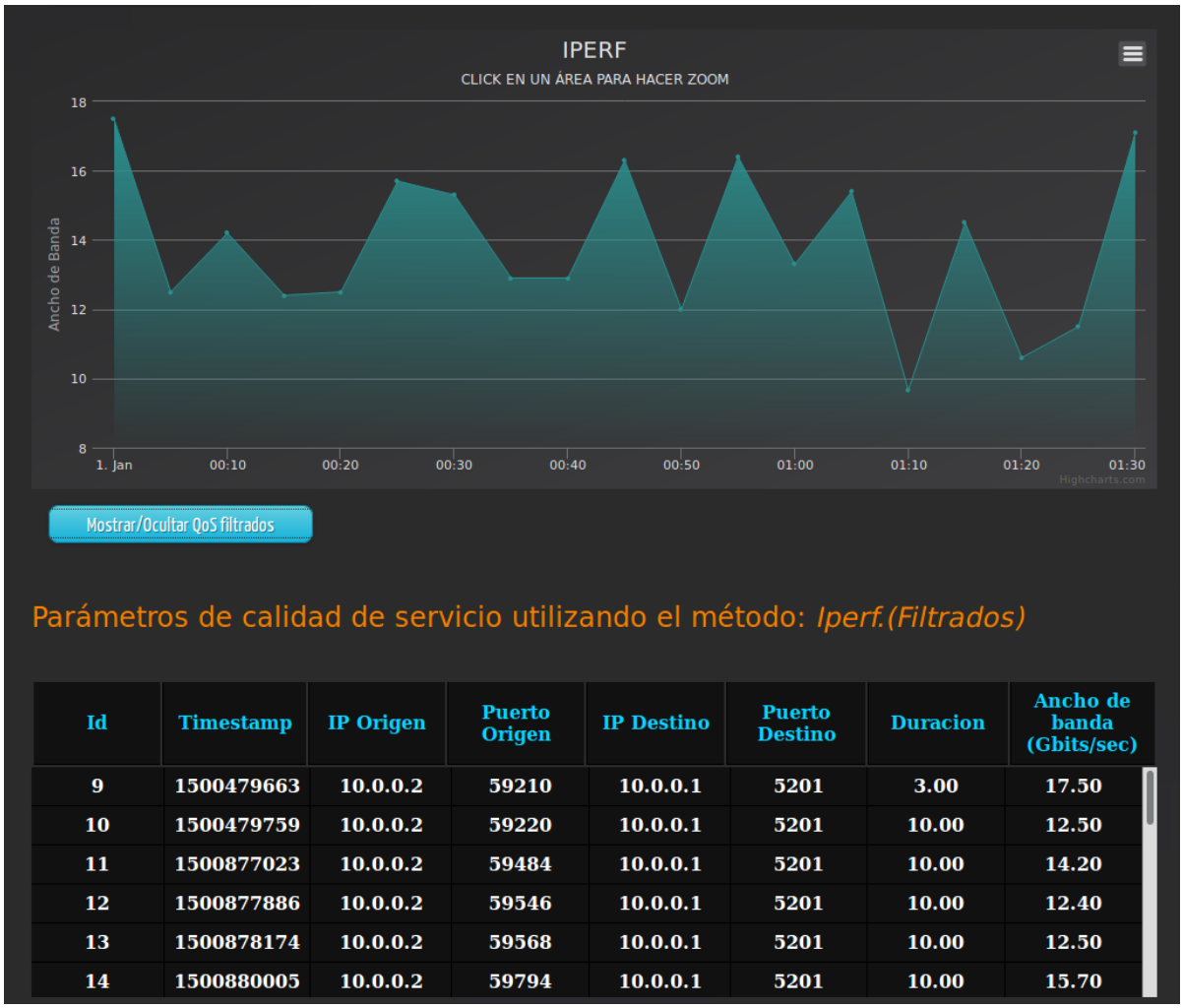



Figura 4.10: Formulario filtro iperf.

También se mostrará un gráfico sobre "*Interval*" o "*Ancho de Banda*" según se desee, a este gráfico también se le aplicará el filtro seleccionado. Para visualizar el gráfico y los valores filtrados se pulsará sobre el botón "*Generar gráfico*". En la figura 4.11 se puede visualizar el resultado.




| Id | Timestamp | IP Origen | Puerto Origen | IP Destino | Puerto Destino | Duracion | Ancho de banda (Gbits/sec) |
|---|---|---|---|---|---|---|---|
| 9 | 1500479663 | 10.0.0.2 | 59210 | 10.0.0.1 | 5201 | 3.00 | 17.50 |
| 10 | 1500479759 | 10.0.0.2 | 59220 | 10.0.0.1 | 5201 | 10.00 | 12.50 |
| 11 | 1500877023 | 10.0.0.2 | 59484 | 10.0.0.1 | 5201 | 10.00 | 14.20 |
| 12 | 1500877886 | 10.0.0.2 | 59546 | 10.0.0.1 | 5201 | 10.00 | 12.40 |
| 13 | 1500878174 | 10.0.0.2 | 59568 | 10.0.0.1 | 5201 | 10.00 | 12.50 |
| 14 | 1500880005 | 10.0.0.2 | 59794 | 10.0.0.1 | 5201 | 10.00 | 15.70 |

Figura 4.11: Resultado filtro iperf.

Si se selecciona la página "*Ping*" se proporciona una definición de la propia herramienta de análisis. También aparecen tres botones de acción: "*Realizar Ping*", "*Realizar filtro*" y "*Mostrar/Ocultar QoS*".


Figura 4.12: Ping.

Si se pincha en la opción "*Realizar ping*" aparece un formulario con tres opciones: "*IP servidor*", "*IP cliente*" y "*Contador*", parámetros necesarios para utilizar el comando ping.


Figura 4.13: Realizar ping.

Una vez rellenados los campos correctamente pulsaremos el botón "*Ejecutar*". Se podrán observar en una fila los parámetros de calidad de servicio más relevantes que se pueden obtener utilizando este método. Se muestra un ejemplo de resultado en la figura 4.14.


Figura 4.14: Resultado ping.

Si se selecciona en el botón "*Mostrar/Ocultar QoS*" podremos consultar el histórico de pings realizados, ver figura 4.15.

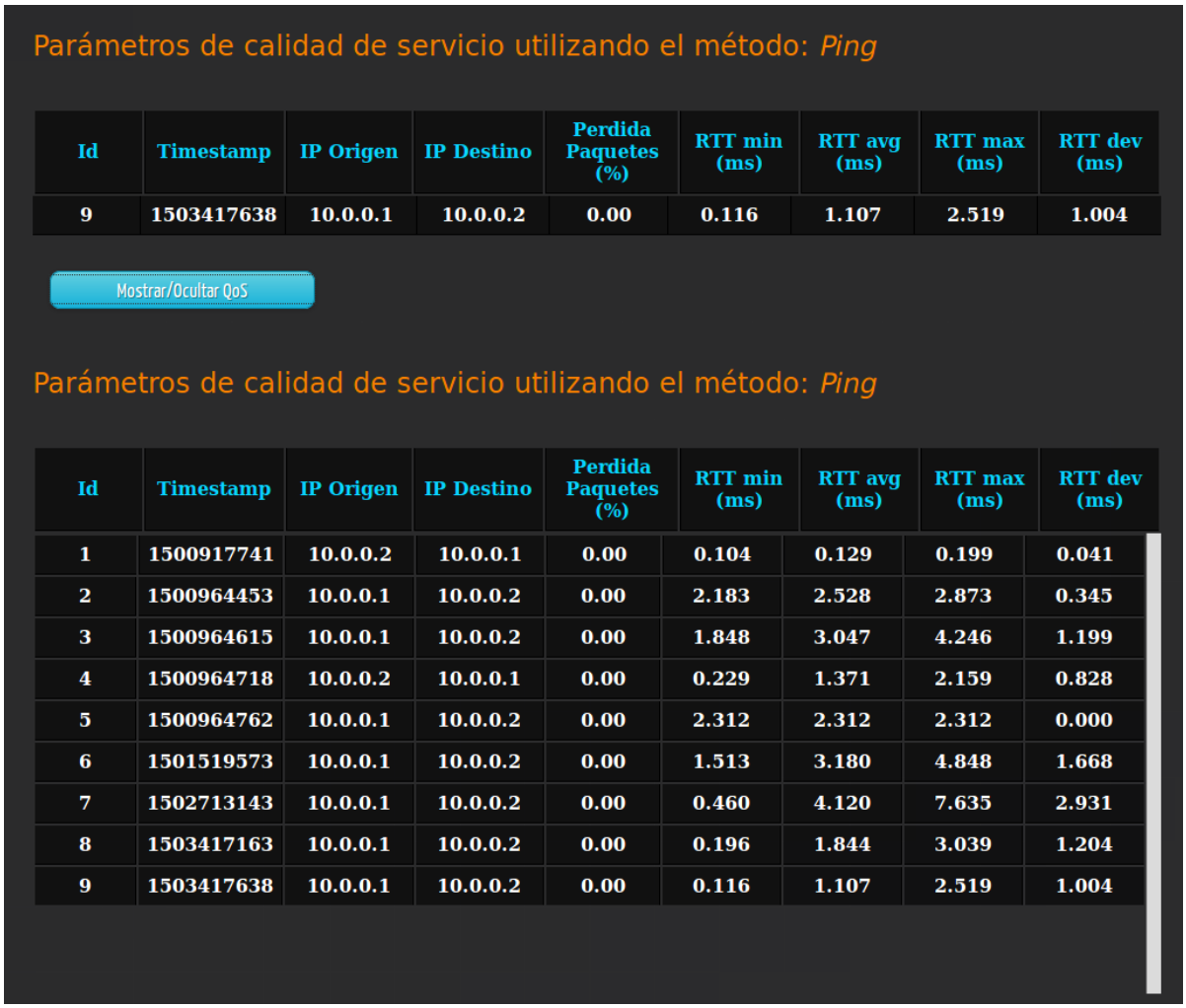



Figura 4.15: Histórico ping.

Si se clica sobre el botón de acción "*Realizar filtro*" podremos obtener gráficas sobre el histórico seleccionando uno de estos campos: "*Paquetes perdidos*", "*RTT min*", "*RTT max*", "*RTT avg*" y "*RTT dev*". Estas gráficas se pueden filtrar por los campos: "*IP origen*", "*IP destino*" y "*RTT min*". Debajo de la gráfica aparecerá un botón para visualizar en forma de tabla los campos filtrados. Ver figura 4.16.

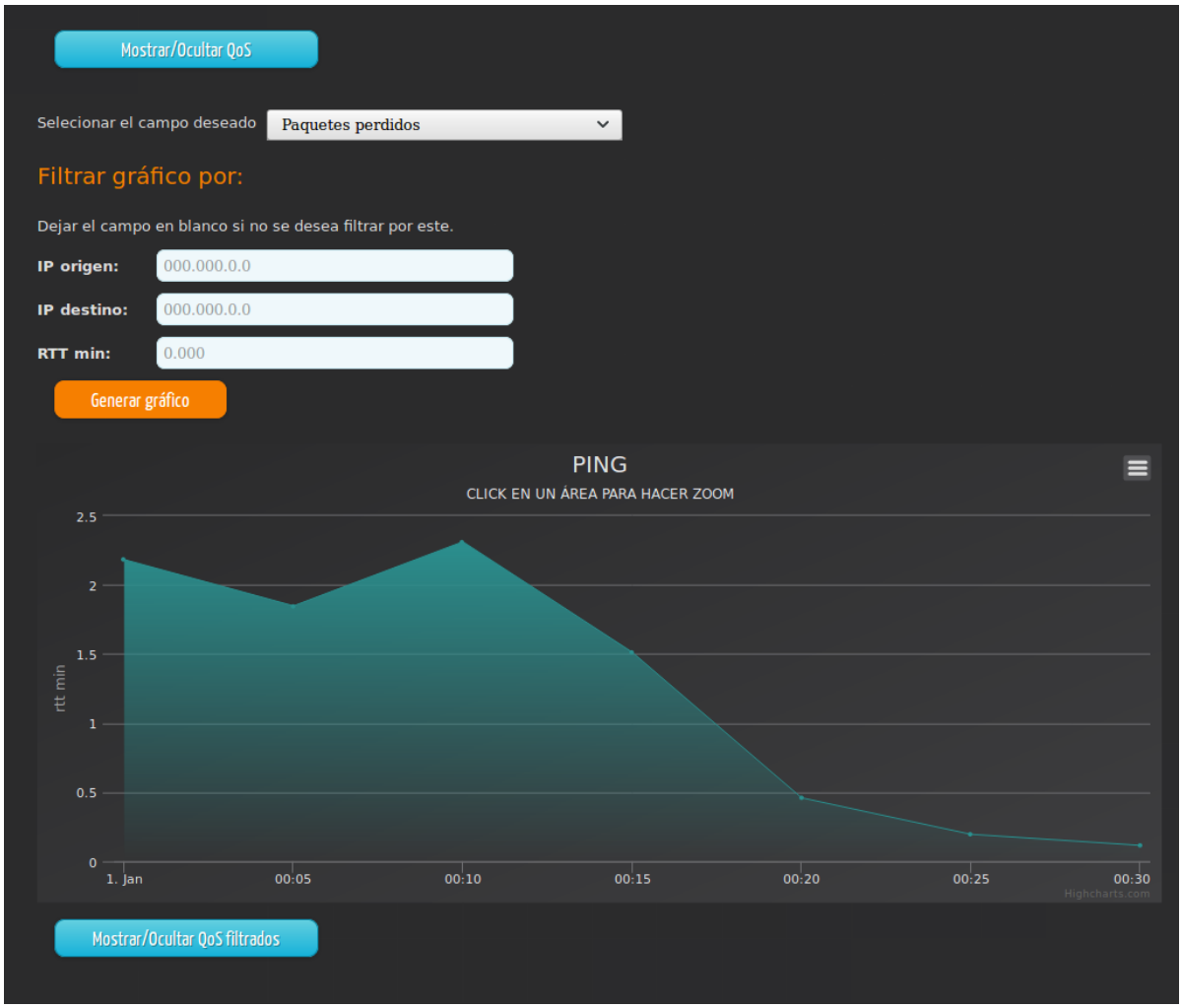
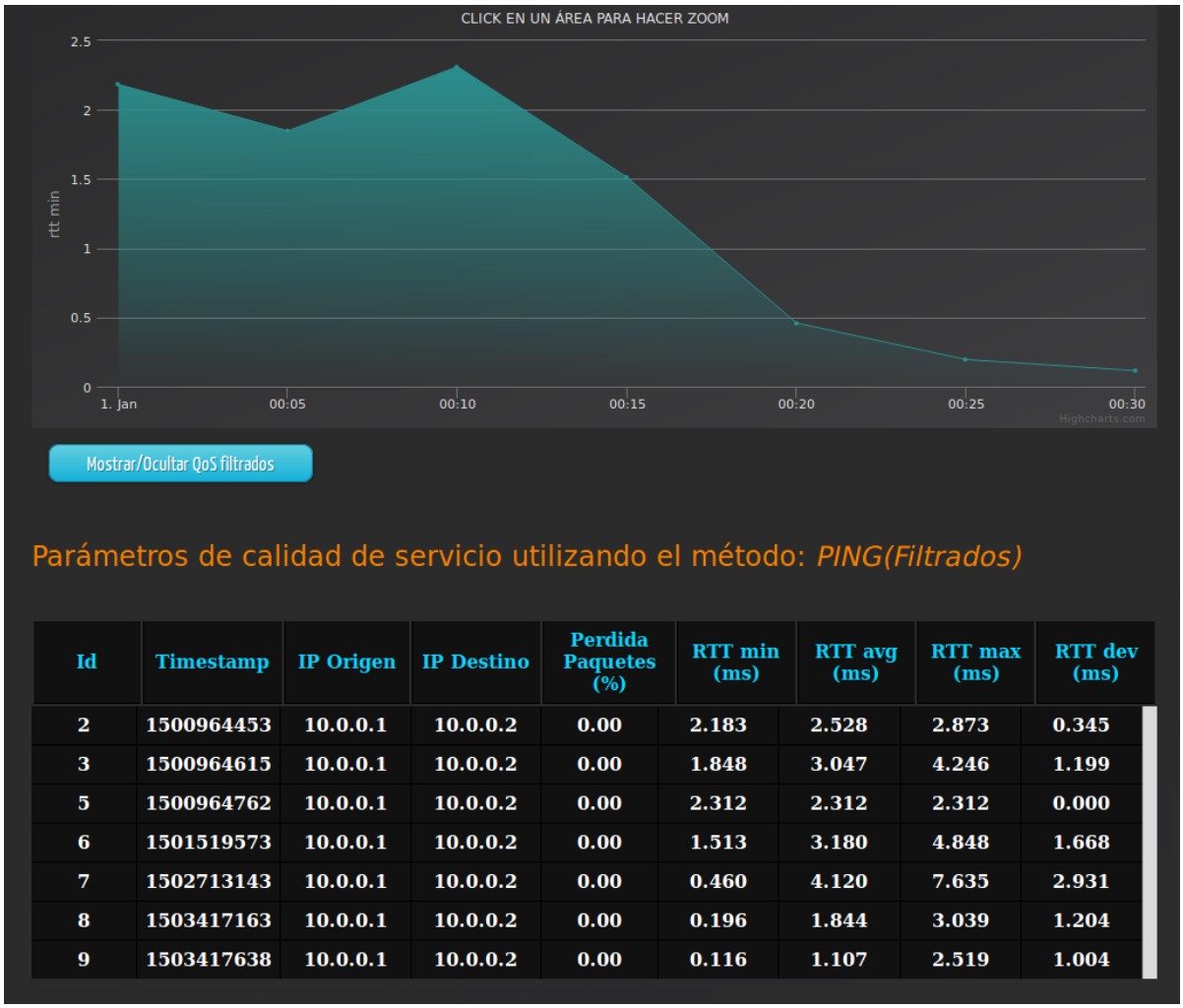



Figura 4.16: Gráfica ping.

Si se selecciona en el botón anteriormente mencionado "*Mostrar/Ocultar QoS filtrados*" aparece el histórico filtrado por los campos que se hayan seleccionado. En este ejemplo se ha filtrado por la IP origen 10.0.0.1, ver figura 4.17.




| Id | Timestamp | IP Origen | IP Destino | Perdida Paquetes (%) | RTT min (ms) | RTT avg (ms) | RTT max (ms) | RTT dev (ms) |
|---|---|---|---|---|---|---|---|---|
| 2 | 1500964453 | 10.0.0.1 | 10.0.0.2 | 0.00 | 2.183 | 2.528 | 2.873 | 0.345 |
| 3 | 1500964615 | 10.0.0.1 | 10.0.0.2 | 0.00 | 1.848 | 3.047 | 4.246 | 1.199 |
| 5 | 1500964762 | 10.0.0.1 | 10.0.0.2 | 0.00 | 2.312 | 2.312 | 2.312 | 0.000 |
| 6 | 1501519573 | 10.0.0.1 | 10.0.0.2 | 0.00 | 1.513 | 3.180 | 4.848 | 1.668 |
| 7 | 1502713143 | 10.0.0.1 | 10.0.0.2 | 0.00 | 0.460 | 4.120 | 7.635 | 2.931 |
| 8 | 1503417163 | 10.0.0.1 | 10.0.0.2 | 0.00 | 0.196 | 1.844 | 3.039 | 1.204 |
| 9 | 1503417638 | 10.0.0.1 | 10.0.0.2 | 0.00 | 0.116 | 1.107 | 2.519 | 1.004 |

Figura 4.17: Resultado filtro ping.

Por último, se encuentra la página "*Fprobe*". Como en las dos anteriores, al acceder obtendremos una breve explicación sobre el funcionamiento de esta herramienta de análisis y cuatro botones de acción:" Capturar *tráfico/Detener servicio*", "*Ver resultado*", "*Realizar filtro*" y "*Mostrar/Ocultar QoS*". Ver figura 4.18.


Figura 4.18: FPROBE.

Si se hace clic en el botón "*Capturar tráfico/Detener servicio*" nos aparecerá un formulario en el que nos pedirá algunos parámetros de configuración. Esto permitirá a la sonda empezar a capturar tráfico NetFlow. El botón "*Detener servicio*" cierra todas las sondas levantadas.


Figura 4.19: FPROBE Capturar/Detener servicio.

Cuando se pulse "*Ejecutar*" aparecerá un mensaje de confirmación indicando que la sonda ha sido activada. Para visualizar los resultados que se están recogiendo pinchamos en el botón de acción "*Ver resultados*" donde introduciremos la ruta que habíamos configurado para guardar los ficheros generados en el paso anterior. La salida de los resultados se puede observar en la figura 4.20.


Figura 4.20: Resultado FPROBE.

Si se selecciona el botón "*Realizar filtro*" podremos obtener una gráfica del "*Número de paquetes*" o el "*Número de bytes*". Esta gráfica se puede filtrar por uno de estos campos: "*IP origen*", "*IP destino*", "*Puerto origen*" y "*Puerto destino*". Ver figura 4.21.

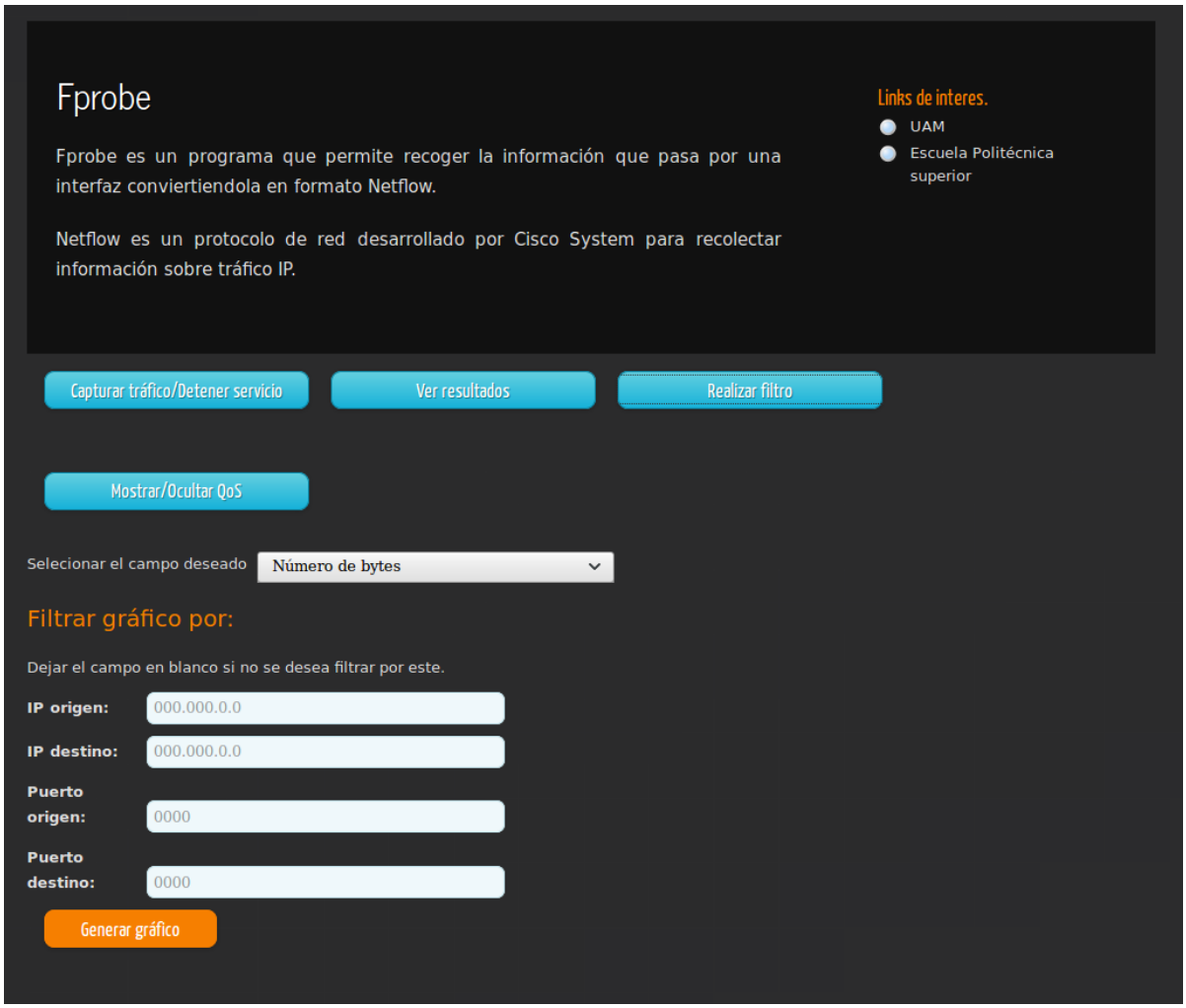



Figura 4.21: Filtro FPROBE.

Si se clica en "*Generar gráfico*" se mostrará la gráfica filtrada como se haya configurado. El resultado de la gráfica se puede ver en la figura 4.22.

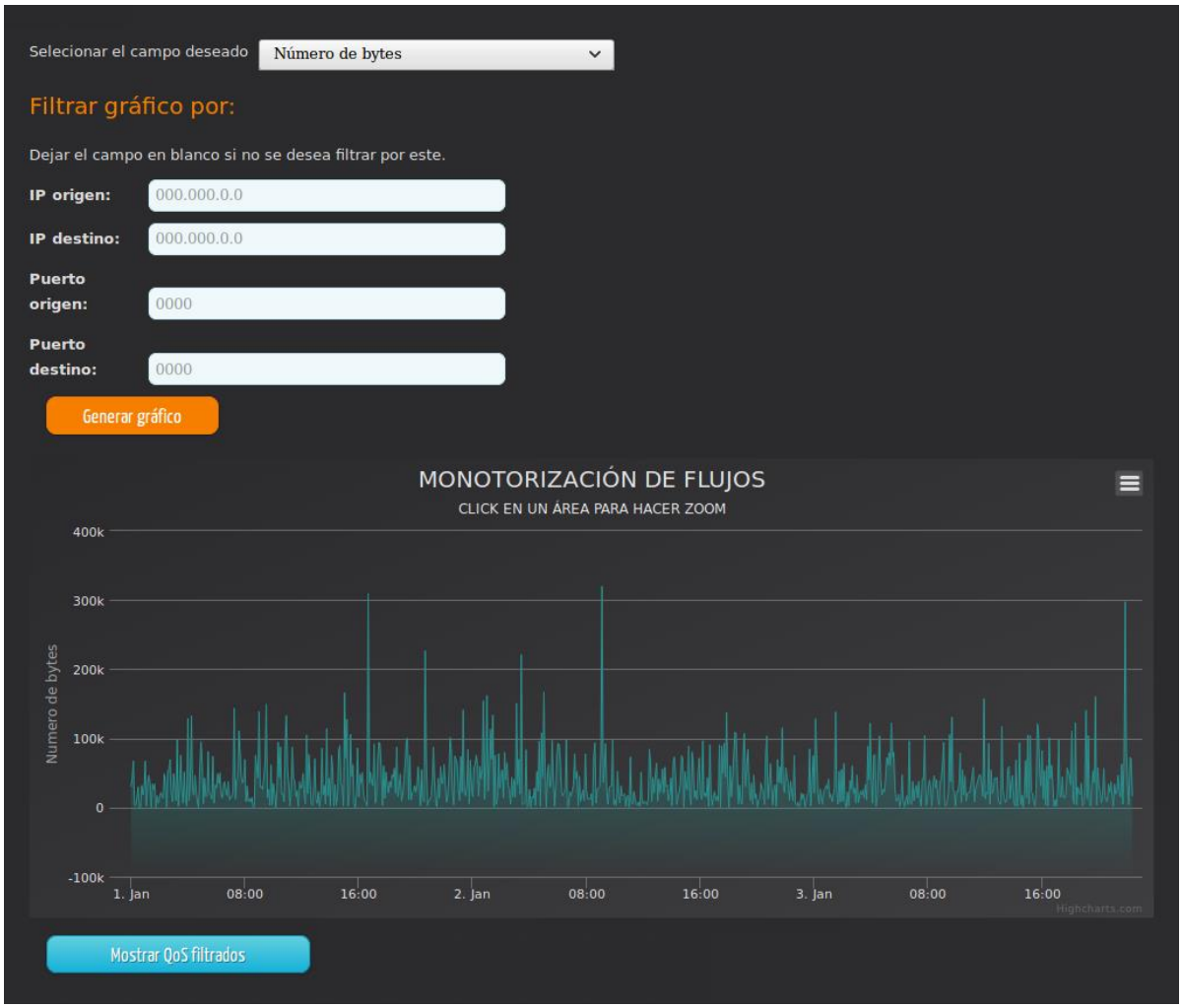
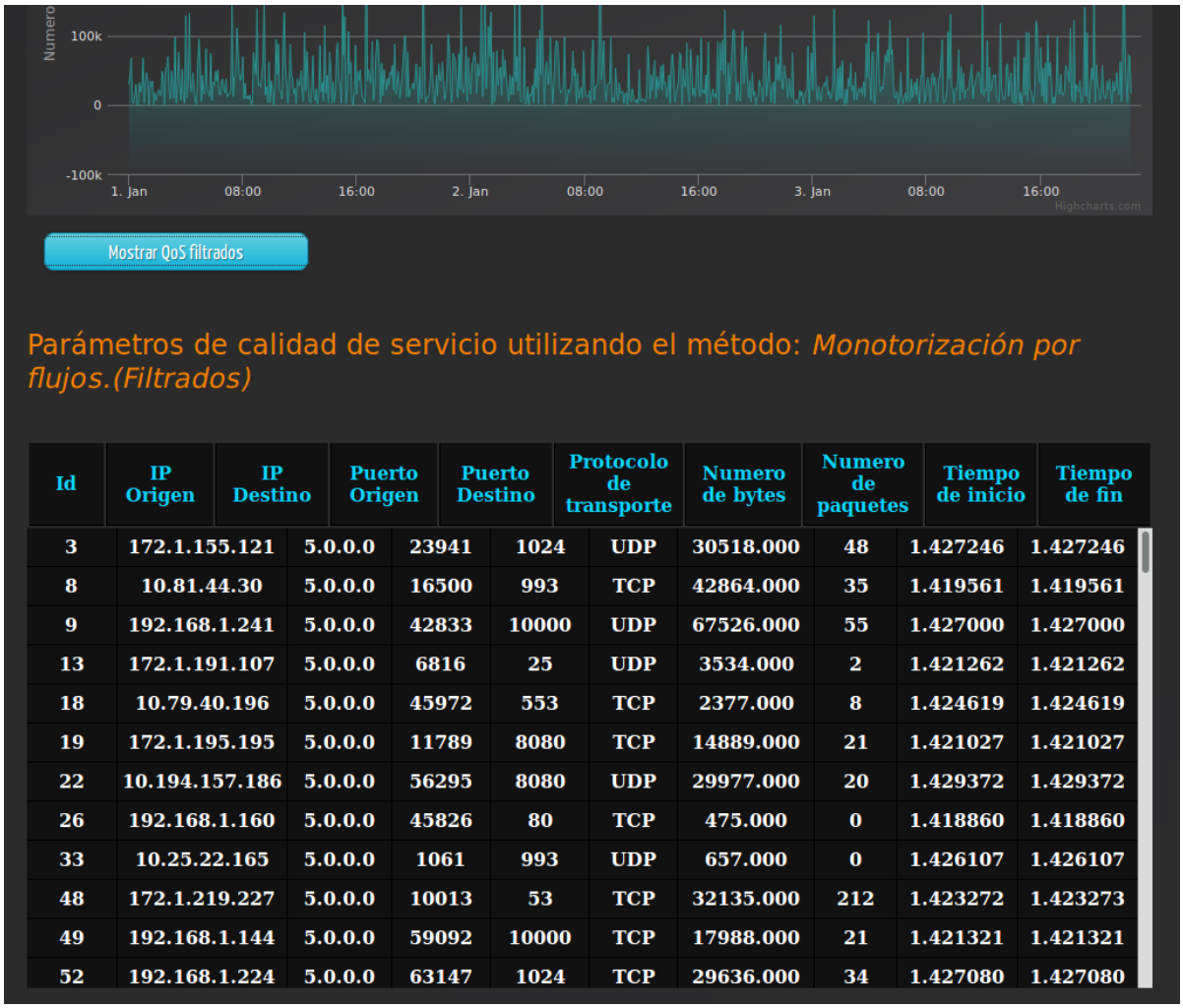



Figura 4.22: Gráfica FPROBE.

Por último, si seleccionamos el botón "*Mostrar QoS filtrados*", podremos ver los parámetros que aparecen en la gráfica. En este ejemplo se ha aplicado el filtro IP destino igual a 5.0.0.0. Ver figura 4.23.




| Id | IP Origen | IP Destino | Puerto Origen | Puerto Destino | Protocolo de transporte | Numero de bytes | Numero de paquetes | Tiempo de inicio | Tiempo de fin |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 172.1.155.121 | 5.0.0.0 | 23941 | 1024 | UDP | 30518.000 | 48 | 1.427246 | 1.427246 |
| 8 | 10.81.44.30 | 5.0.0.0 | 16500 | 993 | TCP | 42864.000 | 35 | 1.419561 | 1.419561 |
| 9 | 192.168.1.241 | 5.0.0.0 | 42833 | 10000 | UDP | 67526.000 | 55 | 1.427000 | 1.427000 |
| 13 | 172.1.191.107 | 5.0.0.0 | 6816 | 25 | UDP | 3534.000 | 2 | 1.421262 | 1.421262 |
| 18 | 10.79.40.196 | 5.0.0.0 | 45972 | 553 | TCP | 2377.000 | 8 | 1.424619 | 1.424619 |
| 19 | 172.1.195.195 | 5.0.0.0 | 11789 | 8080 | TCP | 14889.000 | 21 | 1.421027 | 1.421027 |
| 22 | 10.194.157.186 | 5.0.0.0 | 56295 | 8080 | UDP | 29977.000 | 20 | 1.429372 | 1.429372 |
| 26 | 192.168.1.160 | 5.0.0.0 | 45826 | 80 | TCP | 475.000 | 0 | 1.418860 | 1.418860 |
| 33 | 10.25.22.165 | 5.0.0.0 | 1061 | 993 | UDP | 657.000 | 0 | 1.426107 | 1.426107 |
| 48 | 172.1.219.227 | 5.0.0.0 | 10013 | 53 | TCP | 32135.000 | 212 | 1.423272 | 1.423273 |
| 49 | 192.168.1.144 | 5.0.0.0 | 59092 | 10000 | TCP | 17988.000 | 21 | 1.421321 | 1.421321 |
| 52 | 192.168.1.224 | 5.0.0.0 | 63147 | 1024 | TCP | 29636.000 | 34 | 1.427080 | 1.427080 |

Figura 4.23: Resultado filtro FPROBE.

## *4.2 Aplicaciones NETCONF*

Para cada función definida en el modelo de datos, que se presenta en el Anexo I, se ha desarrollado un script cliente. Cada script cliente llama a su vez a un programa C que realiza una petición NETCONF haciendo uso de la libnetconf. De esta manera si se quieren hacer cambios de funcionalidad no deben más que modificarse los programas C sin tener que cambiar los scripts ni la parte escrita en Django.

Cada programa cliente realizará una petición NETCONF haciendo uso del elemento <rpc> según la funcionalidad del mismo. Cuando la llamada del cliente llega al servidor, este a través del framework transAPI y con la ayuda de lnctool busca el elemento <rpc> en el modelo de datos proporcionado. Si existe en el modelo de datos el framework transAPI llamará a la función callback correspondiente que habrá sido previamente implementada. En la función de callback se ejecutará la herramienta de medida que se haya solicitado y se devolverá la respuesta

Para la implementación de las funciones de callback se hace uso de un fichero que se crea automáticamente utilizando la herramienta lnctool junto con el modelo de datos. Sobre este fichero se realiza la implementación específica de cada RPC. Este fichero C se compila y enlaza en forma de librería dinámica que puede cargar en tiempo de ejecución la librería libnetconf desde una ruta predefinida. La Figura 4.24 muestras las llamadas realizadas entre cada uno de los módulos desarrollados para el control mediante NETCONF.

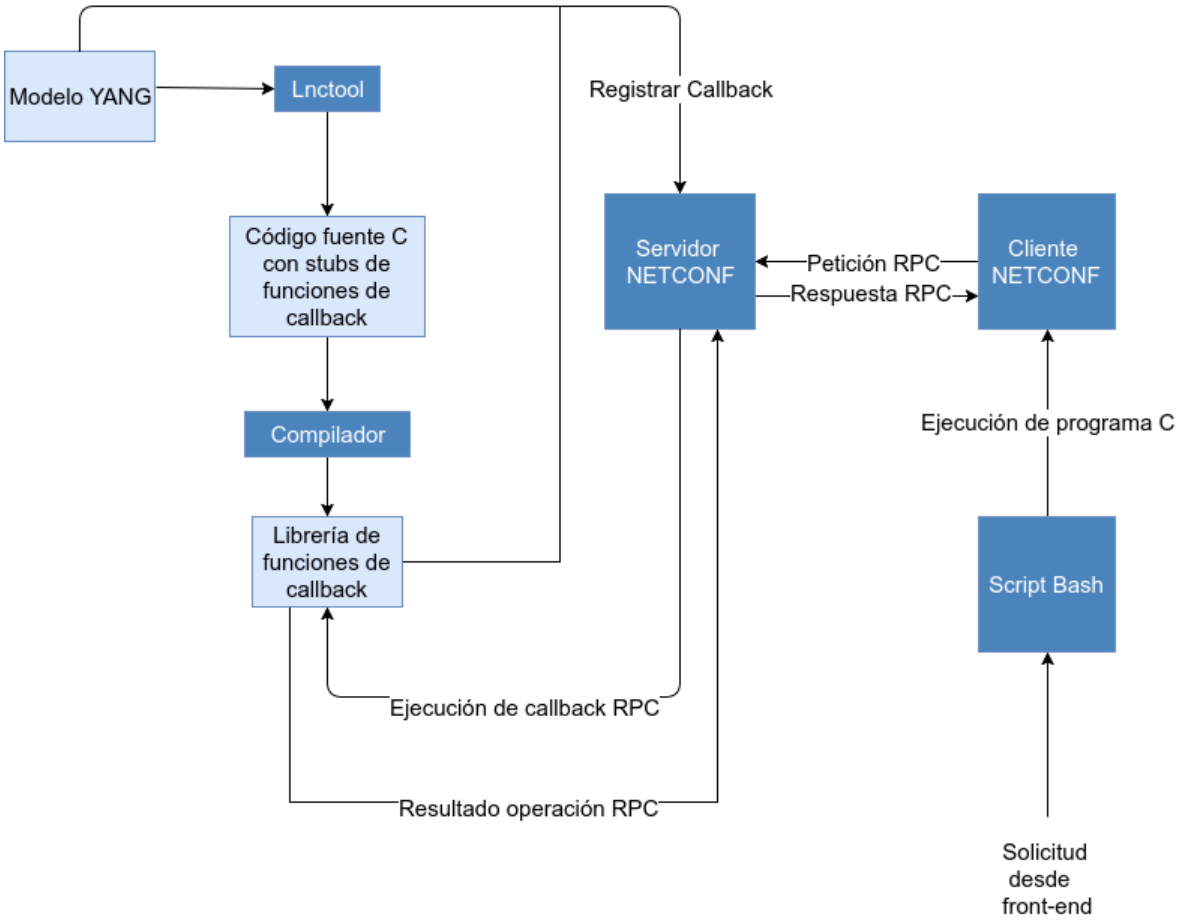



Figura 4.24: Diagrama de llamadas de módulos NETCONF.

## *4.3 Entorno para pruebas*

Para realizar pruebas de configuración y ejecución de herramientas de análisis se ha utilizado mininet [32]. Esta herramienta permite crear redes de computadores virtuales en una máquina individual. Para las pruebas se ejecutó Django dentro de esta red virtual configurada con tres hosts y un switch como se puede ver en la figura 4.25. Se definió un host como el nodo central a partir del cual se definían operaciones de gestión entre los otros dos hosts. Para permitir la conexión SSH entre ambos hosts se creó una clave pública y privada de manera que no fuera necesario introducir la contraseña de sesión cada vez que se realiza una operación.

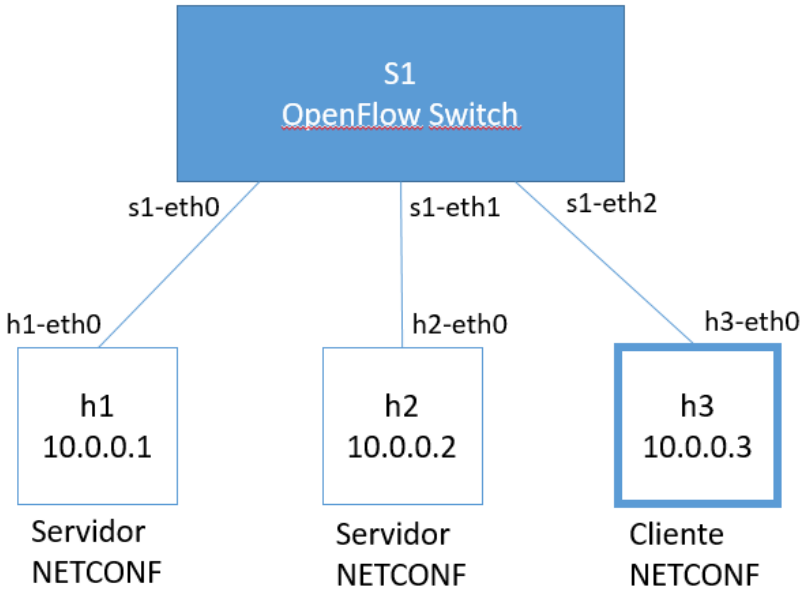



Figura 4.25: Entorno de pruebas.

Para validar el funcionamiento, con la configuración indicada en la figura 4.25, se hicieron las siguientes pruebas:

- Iperf entre h2-h1 configurando h2 servidor y h1 cliente.
- Iperf entre h1-h2 configurando h1 servidor y h2 cliente.
- Ping desde h2 a h1.
- Ping desde h2 a h3.
- Ping desde h1 a h3.
- Ping desde h1 a h2.
- Ping desde h3 a h2.
- Ping desde h3 a h1.
- FPROBE configurando la sonda en h1
- FPROBE configurando la sonda en h2
- Ifconfig de h1
- Ifconfig de h2
- Información hardware h1
- Información hardware h2

# 5 Conclusiones y trabajo futuro

## *5.1 Conclusiones*

La complejidad de Internet aumenta cada día. Esto hace que el protocolo de gestión de red SNMP cada vez sea menos utilizado para gestión y monitorización y motiva la utilización del protocolo NETCONF. A su vez esta complejidad crea la necesidad de poseer una buena herramienta de monitorización y control de las sondas.

El objetivo de este trabajo era el control remoto de sondas Ethernet de bajo coste utilizando un entorno web. Para el control de dichas sondas se decidió hacer uso del protocolo de gestión de red NETCONF y el modelo de datos YANG ya que las tendencias actuales en gestión de redes apuntan en esta dirección. Utilizando estos protocolos se ha creado un sistema basado en un entorno web que permite, de manera intuitiva para el usuario, la ejecución de herramientas y la obtención en tiempo real de parámetros de calidad de servicio.

El resultado del proyecto ha sido un sistema que permite la ejecución remota y en tiempo real de tres herramientas de análisis de red (ping, Iperf y FPROBE) que se ejecutan en las sondas desplegadas y nos permiten obtener una visión amplia del rendimiento de la red. Para ello se han desarrollado varios clientes de NETCONF basados en la librería libnetconf cada uno con una funcionalidad diferente. Además, se han desarrollado modelos YANG que representan diferentes operaciones, así como sus datos de entrada y salida. Se ha desarrollado un servidor NETCONF que permite recibir peticiones de clientes y, a través del framework transAPI ejecutar código específico previamente programado. Los resultados generados por las herramientas de monitorización son devueltos a un sistema web que permite almacenar en una base de datos los parámetros de calidad de servicio (QoS) así como mostrar los resultados. Al usuario que está utilizando la herramienta web solo se le muestran los parámetros de calidad de servicio más relevantes como se ha podido observar en las figuras de la sección 4. A través del entorno web el usuario puede obtener informes sobre el rendimiento de la red.

Este trabajo resulta de gran interés pues la materia que se trabaja está en pleno auge en la actualidad. En primer lugar, está el control remoto de dispositivos utilizado en IoT (Internet of Things) que consiste en conectar cualquier dispositivo electrónico a la red para ser controlado remotamente. En el proyecto también se realiza un entorno web, otra tecnología que está ya consolidada en el mercado y sigue creciendo, pues la mayoría de programas y utilidades que se emplean normalmente están siendo desplazadas a Internet abandonando su habitual uso en local.

Personalmente este proyecto me ha dado la oportunidad de conocer en profundidad el protocolo NETCONF y el modelo de datos YANG, un protocolo simple que te proporciona un gran abanico de posibilidades en cuanto al manejo de dispositivos.

## *5.2 Trabajo futuro*

Como trabajo futuro se abre un mundo de posibilidades, pues una vez obtenido el control de la sonda existen multitud de herramientas que se pueden añadir a las ya mostradas para obtener más información del estado de la red. También se pueden configurar las sondas para que realicen mediciones periódicas y avisen al usuario si se detecta alguna anomalía. Por ejemplo, estableciendo umbrales en los valores medidos. Otra posible línea de trabajo futuro es la creación de un mapa de red con todas las sondas desplegadas de manera que se simplifique la visualización e interacción y se permita la definición de grupos lógicos/físicos de sondas. También se podría configurar el entorno web dando la posibilidad al usuario de levantar y apagar la sonda, lo que resulta útil cuando se tienen levantadas multitud de sondas.

También se deben realizar mejoras en la gestión y reporte de errores, para detectar sondas con errores o malfuncionamiento.

Por otro lado, se propone el estudio de la aplicación de técnicas y sistemas de BigData como Hadoop o Spark para generar sistemas de gestión de sondas que escalen de manera masiva.

Finalmente se deben ir actualizando el entorno web y las funcionalidades del mismo según evolucione la tecnología para que no caiga en desuso y sea una herramienta práctica para los gestores de red.

# Referencias


[1] J. Case, M. Fedor, M. Schoffstall, J. Davin," A Simple Network Management Protocol (SNMP)",MIT Laboratory for Computer Science, May 1990, https://www.ietf.org/rfc/rfc1157.txt.

[2] R. Enns, "NETCONF Configuration Protocol", Juniper Networks, June 2011, https://tools.ietf.org/html/rfc6241.

[3] Tito Cucharero Atienza, "Entorno para la gestión de sondas de red de bajo coste",Universidad Autónoma de Madrid, 2015-06, https://repositorio.uam.es/handle/10486/129795/browse?value=Cucharero+Atienza%2C+Tito&type=author

[4] R. Thurlow, "RPC: Remote Procedure Call Protocol Specification Version 2", Sun Microsystems, May 2009, https://tools.ietf.org/html/rfc5531.

[5] M. Bjorklund, " YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF)",Tail-f System, October 2010, https://tools.ietf.org/html/rfc6020.

[6] J. Rosenberg, "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) ,Cisco, May 2007, https://tools.ietf.org/html/rfc4825.

[7] Javier Ramos. Ph.D. Dissertation, Universidad Autónoma de Madrid, Spain, November 2013.

[8] E. Stephan, RFC 4148: IP Performance Metrics (IPPM) Metrics Registry, August 2005

[9] M. Mathis and M. Allman, RFC 3148: A Framework for Defining Empirical Bulk Transfer Capacity Metrics, 2001.

[10] G. Almes, S. Kalidindi, and M. Zekauskas, RFC 2679: A One-way Delay Metric for IPPM, 1999.

[11] A. Hernandez and E. Magaña, One-way delay measurement and characterization, Proceedings of the 3rd IEEE International Conference on Networking and Services (Athens,Greece), ICNS '07, June 2007, p. 114.

[12] B. Constantine, G. Forget, Ruediger Geib, and R. Schrage, RFC 6349: Framework for TCP Throughput Testing, 2011.

[13] G. Almes, S. Kalidindi, and M. Zekauskas, RFC 2681: A Round-trip Delay Metric for IPPM, 1999.

[14] C. Demichelis and P. Chimento, RFC 3393: IP Packet Delay Variation Metric for IP Performance Metrics (IPPM), 2002.

[15] S. Ickin, K. De Vogeleer, M. Fiedler, and D. Erman, The effects of packet delay variation on the perceptual quality of video, Proceedings of the 35th IEEE Conference on Local Computer Networks, LNC '10, October 2010, pp. 663

[16] G. Almes, S. Kalidindi, and M. Zekauskas, RFC 2680: A One-way Packet Loss Metric for IPPM, 1999.

[17] European Telecommunications Standards Institute, Speech Processing, Transmission and Quality Aspects (STQ); User related QoS parameter definitions and measurements;Part 4: Internet access, 2008.

[18] C. Dovrolis, P. Ramanathan, and D. Moore, What do packet dispersion techniques measure?, Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (Anchorage,Alaska, USA), INFOCOM '01, vol. 2, April 2001, pp. 905-914.

[19] A. Johnsson, On the comparison of packet-pair and packet-train measurements, Proceedings of the 2003 Swedish National Computer Networking Workshop (Arlandastad,Sweden), SNCNW '03, 2003.

[20] B. Melander, M. Bjorkman, and P. Gunningberg, A new end-to-end probing and analysis method for estimating bandwidth bottlenecks, Proceedings of the 2000 IEEE Global Telecommunications Conference (San Francisco, CA, USA), GLOBECOM '00, vol. 1, November 2000, pp. 415-420.

[21] B. Melander, M. Björkman, and P. Gunningberg, Regression based- available bandwidth measurements, Proceedings of the 2002 SCS/IEEE Symposium on Performance and Evaluation of Computer and Telecommunications Systems (San Diego, CA, USA), SPECTS '02, July 2002.

[22] C. Dovrolis, P. Ramanathan, and D. Moore, Packet-dispersion techniques and a capacity-estimation methodology, IEEE/ACM Transactions on Networking 12 (2004), 963-977.

[23] http://pages.cs.wisc.edu/~plonka/FlowScan/INSTALL.html, Accedido el 2 de Junio de 2015

[24] https://capa3.es/medir-el-ancho-de-banda-de-la-red-con-iperf-o-jperf.html, Accedido el 24 de Agosto de 2017

[25] https://nebul4ck.wordpress.com/2015/06/25/generar-recolectar-y-tratar-netflow/, Accedido el 24 de Agosto de 2017

[26] Brian Hedstrom, Akshay Watwe, Siddharth Sakthidharan, Protocol Efficiencies of NETCONF versus SNMP for Configuration Management Functions.University of Colorado, May 2,2011.

[27] M. Bjorklund, YANG – A Data Modeling Language for Network Configuration Protocol (NETCONF), Tail-f Systems, October 2010, https://tools.ietf.org/html/rfc6020

[28] https://github.com/cesnet/libnetconf, Accedido el 24 de Agosto del 2017.

[29] Gerhard Münz, Albert Antony, Falko Dressler, and Georg Carle, Using Netconf for Configuration Monitoring Probes, University of Tübingen, Germany, WSI – Computer Networks & Internet, 2002.

[30] Vaibhav Bajpai,Computer Science, Managing SamKnows Probes using NETCONF, University Bremen.

[31] https://www.highcharts.com/, Accedido el 24 de Agosto del 2017.

[32] http://mininet.org, Accedido el 24 de Agosto del 2017.

# Glosario

**ACK** *Acknowledgement* (Asentimiento), confirmación de un mensaje que fue enviado desde el destino hacia el origen.

**BTC** *Bulk Data Transfer Capacity*, es una característica de aplicación de software que utiliza la compresión de datos.

**CV** *Coefficient of Variation* (Coeficiente de variación).

**DSL** *Digital Subscriber Line* (Line de Abonado Digital), familia de tecnologías que proporciona el acceso a Internet a través de una red telefónica local.

**HTTP** *Hypertext Transfer Protocol* (Protocolo de Transferencia de Hipertexto)**,** protocolo orientado a transacciones en Internet. Sigue el esquema petición-respuesta entre un cliente y un servidor.

**IP** *Internet Protocol* (Protocolo de Internet), protocolo de comunicación clasificado en la capa de red según el modelo OSI.

**ISP** *Internet Service Provider* (Proveedor de Servicios de Internet), empresa que brinda conexión a Internet a sus clientes.

**MRTG** *Multi Router Traffic Grapher* (Grafico de Tráfico Multi-Router).

**MSRP** *Message Session Relay Protocol* (Protocolo de Retransmisión de Sesión de Mensajes), protocolo para transmitir una serie de mensajes instantáneos relacionados en el contexto de una sesión de comunicaciones.

**NIDS** *Network Intrusion Detection System* (Sistema de Detección de Intrusos en una Red), busca detectar anomalías que inicien un riesgo potencial.

**OWD** *One-Way Delay* (Retardo en un Sentido), tiempo necesario para transmitir un paquete a través de una red desde el origen al destino.

**PCAP** *Packet Capture* (Paquete Capturado).

**PRTG** *Paessler Router Traffic Grapher* (Paessler Gráfico del Tráfico de Router), es un software de monitorización de red de Paessler AG.

**QoE** *Quality of Experience* (Calidad de la Experiencia del Usuario).

**QoS** *Quality of Service* (Calidad de Servicio).

**RST** *Reset* (Reseteo), bit que se encuentra en el campo del código en el protocolo TCP, y se utiliza para reiniciar la conexión.

**RTT** *Round-Trip Time* (Retardo de Ida y Vuelta), tiempo requerido para que un paquete viaje a través de una red desde una fuente específica a un destino y vuelva de nuevo.

**SLA** *Service-Level Agreement* (Acuerdo de Nivel de Servicio), contrato en el que se acuerda la calidad de servicio entre un proveedor de servicio y su cliente.

**SDN** *Software Defined Networking* (Redes Definidas por Software), es un conjunto de técnicas cuyo objetivo es facilitar la implementación e implantación de servicios red.

**SNMP** *Simple Network Management Protocol* (Protocolo Simple de Administración de Red), protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos red.

**SPAN** *Switched Port Analyzer* (Analizador de Puertos del Switch), envía copias de paquetes de red vistos en un puerto del switch a una conexión de red monitorizada en otro puerto de switch.

**TCP** *Transmission Control Protocol* (Protocolo de Control de la Transmisión), protocolo de transporte que garantiza que los datos serán entregados sin errores y en el mismo orden en que se transmitieron.

**UDP** *User Datagram Protocol* (Protocolo de Datagramas de Usuario), protocolo del nivel de transporte basado en el intercambio de datagramas.

**VoIP** *Voice over IP* (Voz sobre IP), grupo de recursos que posibilita que viaje la señal de voz a través de Internet, utilizando IP.

**WMI** *Windows Management Instrumentation* (Instrumental de Administración de Windows), es una iniciativa que pretende establecer normas estándar para tener acceso y compartir la información de administración a través de la red de una empresa.

# Anexo I

**Modelo de datos YANG:**

```
module toaster {
  namespace "http://netconfcentral.org/ns/toaster";
  prefix toast;

  container toaster;
  rpc make-toast {
    description
      "Make some toast.
       The toastDone notification will be sent when
       the toast is finished.
       An 'in-use' error will be returned if toast
       is already being made.
       A 'resource-denied' error will be returned
       if the toaster service is disabled.";
  }
  rpc make-ls {
    description
      "ls.";
  }
  rpc iperf_s {
    description
      "Arranca un servidor de iperf";
  }
  rpc iperf_c {
    description
      "Arranca cliente iperf";
  }
  rpc iperf_stop {
    description
      "Detiene servidores iperf";
  }
  rpc fprobe_start {
    description
      "Captura trafico pasivo con la ayuda de fprobe.";
  }
  rpc fprobe_result {
    description
      "Obtiene el trafico capturado.";
  }
  rpc ifconfig {
    description
      "ifconfig";
  }
  rpc general_information {
```

```
    description
      "obtiene informacion general sonda.";
  }
  rpc ping {
    description
      "Realiza el comando ping.";
  }
  rpc fprobe_stop {
    description
      "Detiene fprobe";
  }
}
```